SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING

(Deemed to be University)

Syllabus for

M.Tech.(Computer Science)

(Applicable from the batch 2017–18 onwards)

Prasanthi Nilayam – 515 134

Anantapur District, Andhra Pradesh, Ph: (08555) 287239, Fax: 286919

Website: www.sssihl.edu.in ; Email: registrar@sssihl.edu.in

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING

(Deemed to be University)

Syllabus for M.Tech. (Computer Science)

(Applicable from the batch 2017–18 onwards)

Programme Objectives

1. To prepare graduates who will be successful professionals in industry, government, academia, research, entrepreneurial pursuit and consulting firms
2. To prepare graduates who will contribute to society as broadly educated, expressive, ethical and responsible citizens with proven expertise
3. To prepare graduates who will achieve peer-recognition; as an individual or in a team; through demonstration of good analytical, research, design and implementation skills
4. To prepare graduates who will pursue higher goals of life-not succumb to mundane pressures of life by having clarity in thought, word and deed

Programme specific Outcomes

1. an ability to apply knowledge of mathematics, science and engineering in practice
2. an ability to identify, critically analyze, formulate and solve engineering problems with comprehensive knowledge in the area of specialization
3. an ability to contribute by research and innovation to solve engineering problems
4. an ability to come up with a model and evaluation criteria so that the results/solution can be interpreted to provide for well-informed conclusions
5. an ability to understand the impact of engineering solutions in a contemporary, global, economic, environmental, and societal context for sustainable development
6. an ability to function professionally with ethical responsibility as an individual as well as in multidisciplinary teams with positive attitude

Basic Structure

M.Tech.(Computer Science) is a four-semester programme for 74 Credits. Students with either Bachelor's degree in Engineering or Master's degree in Science are admitted. The students are expected to have undergone formal educational training with appropriate credits in first level courses in Computer organization and architecture, Computer networks, Data base systems, Systems programming. Students are also expected to have undergone formal training in Programming with C, C++ and Java language. These courses constitute the prerequisites for the M.Tech.(Computer Science) programme. In addition to this, fair knowledge of Operating systems, Compiler Design, Formal Languages and Automata Theory, hands on experience with UNIX environment and familiarity in programming with PYTHON, MATLAB etc. will be advantageous to the students seeking admission to this course. A student takes in all 10 courses of which 6 are core courses which pertain to the fundamentals of computer science. The remaining 4 are elective courses which give the scope for specialization in the individual's interest or thrust areas of the department.

Course Design

Semester Duration: A maximum of 14 weeks is typically used in the semester to deliver any subject.

Credit Distribution:

1. Total 74 credits for the course is distributed as given below:

a.	6 Core Theory subjects	18 (3 credits each)
b.	5 Core Practical subjects	10 (2 credits each)
c.	4 Elective subjects	16 (4 credits each)
d.	4 Seminar presentations	4 (1 credit each)
e.	2 Viva voce at the end of semester	2 (1 credit each)
f.	Comprehensive Viva voce in the fourth semester	2 credits
g.	Project during the third and fourth semester	18 credits
h.	Awareness Courses	4 credits

2. An elective subject may be designed to have two different components/mode of delivery namely, Theory and Practical with clear division of credits among the components. For example, a 4 credit course could be specified as 2L+2P to indicate 2 credits for lectures and 2 credits for practical or as 3L+P meaning 3 credits for lecture and one credit for practical or L+3P to indicate 1 credit for lectures and 3 credits for practical.

3. Typically, one lecture credit (T) is given one period per week, one practical credit (P) is given minimum of two and maximum of three periods per week.

4. In the first two semesters students have four seminar presentations that demand independent study of research papers, latest technology trends in the areas of interest and presentation skills. Seminar-I & III will be on study of Research Papers published in journals. Seminar-II & IV will be based on happenings and latest technology trends in the area of Computer Science.

5. The Comprehensive Viva voce is conducted in the final semester.

6. The students have to do a two-semester project starting from third semester.

* * *

DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE

SCHEME OF INSTRUCTION AND EVALUATION
M.Tech.(Computer Science)
(Effective from the batch 2017-18 onwards)

Paper Code	Title of the Paper	Credits	Hours	Types of papers	Modes of Evaluation	Maximum Marks
FIRST SEMESTER						
MTCS-101	Design and Analysis of Algorithms	3	3	T	IE2	100
MTCS-101(P)	Practicals: Design and Analysis of Algorithms	2	4	P	I	50
MTCS-102	Advanced Computer Architecture	3	3	T	IE2	100
MTCS-102(P)	Practicals: Advanced Computer Architecture	2	4	P	I	50
MTCS-103	Parallel Processing	3	3	T	IE2	100
MTCS-103(P)	Practicals: Parallel Processing	2	4	P	I	50
MTCS-104	Elective-I	4*	4*	T	IE2	100*
MTCS-105	Seminar-I	1	1	-	I	50
MTCS-106	Seminar-II	1	1	-	I	50
MTCS-107	Semester End Viva voce	1	-	SEV	E1	50
MAWR-100	Awareness Course-I: Fundamentals of Indian Culture	1	1	T	I	50
		--- 23	**28**			**------ 750**
SECOND SEMESTER						
MTCS-201	Theory of Computation	3	3	T	IE2	100
MTCS-202	Distributed Systems	3	3	T	IE2	100
MTCS-202(P)	Practicals: Distributed Systems	2	4	P	I	50
MTCS-203	Topics in Database Management Systems	3	3	T	IE2	100
MTCS-203(P)	Practicals: Topics in Database Management Systems	2	4	P	I	50
MTCS-204	Elective-II	4*	4*	T	IE2	100*
MTCS-205	Elective-III	4*	4*	T	IE2	100*
MTCS-206	Seminar-III	1	1	-	I	50
MTCS-207	Seminar-IV	1	1	-	I	50
MTCS-208	Semester End Viva voce	1	-	SEV	E1	50
MAWR-200	Awareness Course-II: Sources of Values	1	1	T	I	50
		--- 25*	**28***			**------ 800**
THIRD SEMESTER						
MTCS-301	Elective-IV	4*	4*	T	IE2	100*
MTCS-401	Project Work – Review	-	22	PW	-	50**
MAWR-300	Awareness Course-III: Work Culture, Ethics and Values	1	1	T	I	50
		--- 5	**27***			**------ 200**

FOURTH SEMESTER

MTCS-401	Project Work	18	24	PW	E2	150***
MTCS-402	Comprehensive Viva voce	2	-	COV	E1	50
MAWR-400	Awareness Course-IV: SSSIHL's Core Values and Philosophy	1 -- **21**	1 **25**	T	I	50 -----**250**
	TOTAL:	**74**	**108***			**2000***

PS: Please refer to guidelines for 'Modes of Evaluation for various types of papers', and 'Viva voce nomenclature & scope and constitution of the Viva voce Boards'.

* Credits split between Lectures and Practical, total marks for the subject, and the grand total marks for the paper, may change based on the credits allocated for the Lecture and Practicals of the elective(s) the students opt for. i.e., the elective paper A may be designed to have two different components/mode of delivery namely, Theory and Practical with clear division of credits among the components. For example, a 4 credit course could be specified as 2L+2P to indicate 2 credits for lectures and 2 credits for practical or as 3L+P meaning 3 credits for lecture and one credit for practical or L+3P to indicate 1 credit for lectures and 3 credits for practical.

** The Project Work topic would be finalized by the end of the second semester, and the Project Work starts in the third semester and gets completed in the fourth semester. The interim review would consist of an oral examination to assess the progress made by the student in the project work. Students will be asked to make a presentation along with a submission of report of work done so far.

*** Total marks for the Project Work would be for **200 marks**, which would include

- **50 marks** for the review of the project work by the student at the end of the third semester (please see **)
- **100 marks** for the Project Report Examination
- **50 marks** for Project Viva voce conducted at the end of the 4th semester.

Continuous Internal Evaluation (CIE) & End Semester Examination (ESE)

Indicator	**Legend**
IE1	CIE and ESE ; ESE single evaluation
IE2	CIE and ESE ; ESE double evaluation
I	Continuous Internal Evaluation **(CIE)** only Note: 'I' does not connote 'Internal Examiner'
E	End Semester Examination **(ESE)** only Note: 'E' does not connote 'External Examiner'
E1	ESE single evaluation
E2	ESE double evaluation

STREAMS of Elective Courses

STREAM I: INTELLIGENT SYSTEMS AND KNOWLEDGE ENGINEERING

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
ISKE 1	Artificial Intelligence	T	3	--
ISKE 1(P)	Practicals: Artificial Intelligence	P	1	
ISKE 2	Genetic Algorithms	T	3	--
ISKE 2(P)	Genetic Algorithms	P	1	
ISKE 3	Natural Language Processing	T	3	First level Course in A.I
ISKE 3(P)	Practicals: Natural Language Processing	P	1	First level Course in A.I
ISKE 4	Neural Networks	T	3	--
ISKE 4(P)	Practicals: Neural Networks	P	1	--
ISKE 5	Data Mining and Data Warehousing	T	3	First level course in Databases
ISKE 5(P)	Practicals: Data Mining and Data Warehousing	P	1	First level course in Databases
ISKE 6	Pattern Recognition	T	3	Foundations in Probability and Statistics.
ISKE 6(P)	Practicals: Pattern Recognition	P	1	Foundations in Probability and Statistics.
ISKE 7	Machine Learning	T	3	Foundations in Probability and Statistics.
ISKE 7(P)	Practicals: Machine Learning	P	1	Foundations in Probability and Statistics.
ISKE 8	Mining of Big Data Sets	T	2	--
ISKE 8(P)	Practicals: Mining of Big Data Sets	P	2	--
ISKE 9	Deep Learning	T	2	--
ISKE 9(P)	Practicals: Deep Learning	P	2	--

STREAM II: ADVANCED COMPUTER NETWORKS

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
ACN 1	Telecom Networking	T	3	First level Course in Computer Networks
ACN 1(P)	Practicals: Telecom Networking	P	1	First level Course in Computer Networks
ACN 2	Network Security	T	3	First level Course in Computer Networks
ACN 2(P)	Practicals: Network Security	P	1	
ACN 3	Wireless and Mobile Networks	T	3	First level Course in Computer Networks
ACN 3(P)	Practicals: Wireless and Mobile Networks	P	1	First level Course in Computer Networks
ACN 4	Advanced Computer Networks	T	3	First level Course in Computer Networks
ACN 4(P)	Practicals: Advanced Computer Networks	P	1	First level Course in Computer Networks

STREAM III: HUMAN COMPUTER INTERACTION

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
HCI 1	Digital Image Processing	T	3	--
HCI 1(P)	Practicals: Digital Image Processing	P	1	
HCI 2	Medical Image Processing	T	3	First level Course in Image Processing

HCI 2(P)	Practicals: Medical Image Processing	P	1	First level Course in Image Processing
HCI 3	Computer vision	T	3	First level Course in Image Processing
HCI 3(P)	Practicals: Computer vision	P	1	First level Course in Image Processing
HCI 4	Advanced Topics in Image Processing	T	3	First level Course in Image Processing
HCI 4(P)	Practicals: Advanced Topics in Image Processing	P	1	First level Course in Image Processing
HCI 5	Video Processing	T	3	First level Course in Image Processing
HCI 5(P)	Practicals: Video Processing	P	1	First level Course in Image Processing

STREAM IV: THEORETICAL COMPUTER SCIENCE

Paper Code	***Elective Title***	**Theory/ Practical**	**Credits**	***Prerequisite***
TCS 1	Advanced Algorithms	T	3	First level Course in Algorithms, Probability.
TCS 1(P)	Practicals: Advanced Algorithms	P	1	First level Course in Algorithms, Probability.
TCS 2	Cryptography	T	3	Basic Number theory
TCS 2(P)	Practicals: Cryptography	P	1	good skills in programming

STREAM V: COMPUTER SYSTEMS

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
CS 1	Compiler Design	T	3	
CS 1(P)	Practicals: Compiler Design	P	1	
CS 2	Embedded Computing	T	3	First course in Architecture, OS
CS 2(P)	Practicals: Embedded Computing	P	1	First course in Architecture, OS
CS 3	Advanced Programming in the Unix Environment	T	2	
CS 3(P)	Practicals: Advanced Programming in Unix Environment	P	2	
CS 4	Programming for performance	T	2	First course in Architecture
CS 4(P)	Practicals: Programming for Performance	P	2	First course in Architecture
CS 5	Operating Systems	T	3	
CS 5(P)	Practicals: Operating Systems	P	1	

STREAM VI: MULTI-CORE AND PARALLEL COMPUTING

Paper Code	Elective Title	Theory/ Practical	Credits	Prerequisite
MPC 1	Parallel Numerical Linear Algebra	T	3	First level Course in Architecture, Algorithms, Numerical Linear Algebra
MPC 1(P)	Practicals: Parallel Numerical Linear Algebra	P	1	First level Course in Architecture, Algorithms, Numerical Linear Algebra

MPC 2	Multi core Computing	T	3	First level Course in Architecture, Algorithms
MPC 2(P)	Practicals: Multi core Computing	P	1	First level Course in Architecture, Algorithms
MPC 3	High Performance Embedded Computing	T	3	First level Course in Architecture, Algorithms, Operating Systems
MPC 3(P)	Practicals: High Performance Embedded Computing	P	1	First level Course in Architecture, Algorithms, Operating Systems
MPC 4	High Performance Computing with Accelerators	T	2	First Level course in Architecture, Systems Programming
MPC 4(P)	Practicals: High Performance Computing with Accelerators	P	2	First Level course in Architecture, Systems Programming
MPC 5	Cloud Computing	T	2	First Level course in Architecture, Systems Programming
MPC 5(P)	Practicals: Cloud Computing	P	2	First Level course in Architecture, Systems Programming
MPC 6	Multi Processor Programming	T	2	First Level course in Architecture, Systems Programming
MPC 6(P)	Practicals: Multi Processor Programming	P	2	First Level course in Architecture, Systems Programming

STREAM VII: SOFTWARE ENGINEERING

Paper Code	*Elective Title*	**Theory/ Practical**	**Credits**	*Prerequisite*
SE 1	Object Oriented System Design	T	3	--
SE 1(P)	Practicals: Object Oriented System Design	P	1	--
SE 2	Web Technology	T	3	--
SE 2(P)	Practicals: Web Technology	P	1	--

STREAM VIII: MATHEMATICAL METHODS IN COMPUTER SCIENCE

Paper Code	*Elective Title*	*Theory/ Practical*	*Credit s*	*Prerequisite*
MMCS 1	Mathematical Methods in Image Processing	T	3	First level course in PDE and Calculus of Variations
MMCS 1(P)	Practicals: Mathematical Methods in Image Processing	P	1	First level course in PDE and Calculus of Variations
MMCS 2	Numerical Methods in Image Processing	T	3	First level course in Calculus of Variations
MMCS 2(P)	Practicals: Numerical Methods in Image Processing	P	1	First level course in Calculus of Variations
MMCS 3	Mathematical Methods for Data Mining	T	3	First level courses in Probability and Linear Algebra
MMCS 3(P)	Practicals: Mathematical Methods for Data Mining	P	1	First level courses in Probability and Linear Algebra

* * *

M.TECH. (COMPUTER SCIENCE)
CORE COURSES (4 CREDITS)

MTCS-101 DESIGN AND ANALYSIS OF ALGORITHMS

(3 Credits) (42 Periods)

Course Objectives:
The course will familiarize the students with tools for designing and analyzing algorithms. Assuming that the student has studied a basic course in algorithm, the course builds over deeper topics such as proof of correctness of algorithms, design through paradigms and probabilistic analysis. The course provides a broad range of algorithms relevant for the current trends.

Course Outcomes: At the completion of the course the student will be able to

- Know how to solve a problem and subsequently design an algorithm through an inductive process.
- Understand loop invariants and use them to prove the correctness of algorithms.
- Understand different paradigms for designing algorithms such as Divide and Conquer, Dynamic Programming and Greedy Method.
- Understand probabilistic and amortized analysis of algorithms
- Design randomized algorithms
- Know the analysis of various sorting algorithms
- Design solutions for graph problems and prove their correctness.
- Familiarize with specialized topics such as string matching and computational geometry.

Course Syllabus:

Unit 1: Design & Analysis Techniques (10 periods)
Growth of functions, Loop invariant, Divide and Conquer, Master method for solving recurrences, Dynamic programming, Greedy Algorithms (excluding starred sections), Amortized analysis, Probabilistic Analysis and Randomized Algorithms (excluding starred sections).

Unit 2: Sorting & Order statistics (6 periods)
Sorting in linear time, Heapsort, Quicksort, Medians and Order statistics

Unit 3: Data Structures
Elementary data structures, Hash tables, Binary search trees, Red Black trees, Augmenting Data Structure, Data structure for disjoint sets (12 periods)

Unit 4: Graph algorithms (10 periods)
Graph searching techniques, Minimum spanning trees, single source shortest paths, all pairs shortest paths

Unit 5: Specialized topics (4 periods)
String Matching, Computational geometry.

Total Periods: **42** periods

Reference Text: Thomas H Cormen, Charles E Leiserson, Ronald Rivest, Cliffor Stein., Introduction to algorithms, 3rd edition, Chapters: 1 to 17, 21 to 25, 32, 33.

Suggested Reading: Anany Levitin., Introduction to Design & Analysis of Algorithms, 2nd edition.

MTCS-101(P) Practicals: DESIGN AND ANALYSIS OF ALGORITHMS

(2 Credits)

Course Objectives:
Algorithms / Exercises from different units in the syllabus will be implemented in Lab. The student writes their programs in Python language.

Course Outcomes: At the completion of the lab course the student will be
- equipped with the skill set to prove the correctness through strategies such as loop invariants and bound functions.
- Able to write programs by the principles of algorithmic design.

Recommended Exercises:
- convert a recursive programme to an iterative programme
- write programs for various paradigms such as Divide and Conquer, Dynamic Programming and Greedy Method.
- analyze randomized algorithms
- code various sorting algorithms
- work on graph with both representations: adjacency matrix and list
- write code for various graph algorithms
- write code for geometric and string algorithms

Reference Text: Thomas H Cormen, Charles E Leiserson, Ronald Rivest, Cliffor Stein., Introduction to algorithms, 3rd edition, Chapters: 1 to 17, 21 to 25, 32, 33

Suggested Reading: Anany Levitin, Introduction to design & analysis of algorithms, 2nd edition.

* * *

MTCS-102 Advanced Computer Architecture (3 Credits) (42 Periods)

Course Objectives: This course will introduce students to the advanced concepts of computer organization and architecture. It is to familiarize students about design principles of organization and micro-architecture that gives high-performance capability. The course also gives a broader perspective of modern processor designs such as super scalars, vector architecture, GPUs etc.

Course Outcomes: At the completion of the course the student will be able to

- know the classes of computers, and new trends and developments in computer architecture
- Understand pipelining, instruction set architectures, memory addressing.
- Understand the various techniques to enhance a processors ability to exploit Instruction-level parallelism (ILP), and its challenges.
- Understand exploiting ILP using dynamic scheduling, multiple issue, and speculation.
- Understand the performance and efficiency in advanced multiple-issue processors.
- Understand symmetric shared-memory architectures and their performance.
- Understand multiprocessor cache coherence using the directory based and snooping class of protocols.
- Understand the several advanced optimizations to achieve cache performance.
- Understand virtual memory and virtual machines

Course Syllabus:

Unit 1: (7 periods)
Instruction Set Architectures, Microcode (H&P5 Chapter 1, H&P5 Appendix A)
Pipelining Review (H&P5 Appendix C) Cache Review (H&P5 Appendix B-1 - B-40)

Unit 2: (12 periods)
Instruction level parallelism – hardware and software techniques (e.g., dynamic scheduling, superscalar, static and dynamic branch prediction, VLIW, loop unrolling). (H&P5 Chapter 3)
Exceptions (H&P5 Appendix C)

Unit 3: (14 periods)
Branch Prediction (H&P5 Appendix C, Chapter 2) Memory hierarchy – advanced concepts in caches (e.g., prefetching, lockup-free caches, and multi-level caches),
main memory, and virtual memory. (H&P5 Chapter 2 (P.71 – P.105))
Memory Protection (H&P5 Appendix B(B41 -B67), Chapter 2 (P.105 – P.112))

Unit 4: (9 periods)
Vector Processors and GPUs (H&P5 Chapter 4) Multiprocessors/multicore – overview of different models, cache coherence with shared-memory systems/multicore (snoopy and

directory solutions), synchronization. (H&P5 Chapter 5)

Total 42 Periods

Reference text:

1. [H&P5] "Computer Architecture: A Quantitative Approach (5th Edition)", 2012, John L. Hennessy and David A. Patterson, ISBN: 978-0123838728

Suggested readings:

1. "Digital Design and Computer Architecture, 2nd edition" by D. M. Harris and S. L. Harris (Morgan Kaufmann, 2012).

2. "Modern Processor Design: Fundamentals of Superscalar Processors (1st Edition)", 2004. John P. Shen and Mikko H. Lipasti, ISBN: 0070570647, Princeton University Library Owns.

MTCS-102(P) Practicals: Advanced Computer Architecture (2 Credits)

Course Objectives: This lab course is expected to reinforce some of the important concepts and principles that are explained in MTCS-102. main objective is to give hands on training to check and verify some principles taught as part of theory.

Course Outcomes: At the completion of the course the student will be able to

- Install and run a simulator on a computer
- Understand the importance of simulation studies
- Understand various parameters such as instruction cycles, cycles per instruction, execution time etc for a specific benchmark/application program
- Understand the impact of cache configuration on variety of benchmarks
- Introduce new modules into the simulator to rebuild it with certain features pertaining to memory subsystem or CPU etc.

Recommended assignments:

1. Installing and getting acquainted with a computer system simulator such as Gem5
2. Running standard benchmark sets such as SPEC2000 and observing execution characteristics
3. Altering cache configurations to get the best performance for a certain benchmark/application program
4. Introducing a new module in the simulator for memory subsystem to introduce L3 cache
5. Introducing a new module in the simulator for designing multi-core CPU

Reference Material:

1. Gem5 Tutorials and Exercises Online.

* * *

MTCS-103 PARALLEL PROCESSING

(3 Credits) (42 Periods)

Course Objective: To introduce and develop understanding on the following topics:

- Parallelism, Parallel Computers and Parallel Programming
- Parallel Architecture, Interconnection Network and Classification. Distributed Memory and Shared Memory programming using MPI & OpenMP.
- Parallel Algorithms Design, Foster's Design Methodology
- Performance Analysis of Parallel Algorithms and Scalability

Course Outcome: Students will have comprehensive knowledge and develop basic skill on the following concepts:

- Types of Concurrency, recognizing Data Parallelism and Functional Parallelism
- Classification of Architecture
- Properties and classification of Interconnection Network Topologies
- Designing of parallel Algorithm using Foster's Design Methodology and analysis of it.
- Performance Analysis with Amdhal's Law, Gustaffson Barsis's Law, Karp Flat Metric
- Scalability of parallel systems and Iso-efficiency Analysis
- MPI Collective Calls, Barrier synchronization, blocking and Non-blocking Point to Point calls,
- Shared Memory Programming, Parallel Pragmas, number of threads in one program, race conditions, data & functional parallelism in OpenMP.

Course Syllabus:

Unit 0: Introduction - Modern Parallel Computers - Types of Concurrency – Programming. 3 Periods

Unit 1: Parallel Architecture – Interconnection Network – Processor arrays – Multiprocessors – Multi Computers – Flynn's taxonomy. 5 Periods.

Unit 2: Parallel Algorithm Design – Foster's Design Methodology – Example Problems. 4 Periods

Unit 3: Message Passing programming Model – MPI – Point to Point & Collective Calls. 4 Periods.

Unit 4: Algorithms for Illustrations – Sieve of Eratosthenes – Floyd's Algorithm. (To discuss all the concepts introduced so far). 4 Periods

Unit 5: Performance analysis – Speed up and Efficiency – Amdahl's Law – Gustafson's Barsis Law – Karp Flatt Metric – Isoefficiency Metric. 4 Periods

Unit 6: Matrix Vector Multiplication – Monte Carlo Methods – Matrix Multiplication – Solving linear System - finite Difference Methods - sorting algorithm - combinatorial Search. 14 Periods

Unit 7: Shared Memory Programming – Open MP. 4 Periods.

Total: 42 Periods.

Reference Text:
Parallel Programming in C with MPI and OpenMP by Michale J Quinn, Tata McGraw Hill 2004.

Suggested Readings:
Anantha Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Pearson education LPE, 2nd Edition, 2004.

MTCS-103(P) Practicals: PARALLEL PROCESSING (2 Credits)

Course Objectives: To give practical knowledge in:

Shared Memory programming using MPI & OpenMP.
Parallel Algorithms Design
Performance Analysis of Parallel Algorithms and Scalability

Course Outcome: At the completion of the course a student will be sufficiently skilled in parallelizing an algorithm/application using standard message passing libraries such as MPI, OpenMP.

1. Message Passing programming Model – MPI – Point to Point & Collective Calls.
2. Document classification Problem
3. Matrix Vector & Matrix Matrix Multiplication
4. Parallel Quick Sort
5. Shared Memory Programming – Open MP

Reference Text:
Parallel Programming in C with MPI and OpenMP by Michale J Quinn, Tata McGraw Hill 2004.

Suggested Readings:
Introduction to Parallel Computing by Anantha Grama, Anshul Gupta, George Karypis, Vipin Kumar, Pearson education LPE, Second edition, 2004.

* * *

MTCS-201 Theory of Computation

(3 Credits) (42 Periods)

Course Objectives:

- Learn several formal mathematical models of computation along with their relationships with formal languages.
- Learn that not all problems are solvable by computers, and some problems do not admit efficient algorithms.
- Learn techniques to show that a problem is not efficiently solvable

Course Outcomes:

- Demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages.
- Understand that there are limitations on what computers can do, and learn examples of unsolvable problems.
- Appreciate that certain problems do not admit efficient algorithms, and identify such problems.

Course Syllabus:

Unit 1 Introduction to Basic Models of Computation and the finite representation of Infinite Objects 4 periods

Unit 2 Finite Automata and Regular Languages 6 periods

Unit 3 Pushdown Automata and Context - Free Language 6 periods

Unit 4 Turing Machines and their Variants - Recursive Functions - Church's Thesis 14 periods

Unit 5 Un-decidability - Reducibility and Completeness – Time Complexity and NP-Completeness 12 periods

Total 42 periods

Reference Text:

Harry Lewis R, Christos H. Papadimitriou, Elements of theory of computation, 2nd edition, PHI Publications, 1998.

Coverage of Key Text Chapters: 1 (only 1.7 and 1.8), 2, 3, 4 (except 4.4 and 4.6), 5, 6, 7.

Suggested Readings:

John. C. Martin, Introduction to Languages and the Theory of Computation, Tata McGraw-Hill, 2003.

* * *

MTCS-202 DISTRIBUTED SYSTEMS

(3 Credits) (42 Periods)

Course Objectives:

This course deals with the basics, concepts and theory and applications of Distributed Systems. It reviews the fundamentals of Communications, Networking, Internet, World Wide Web. Operating Systems. The subject then moves on to remote procedure calls, name services infrastructure.

The Processes and related issues like Time and Global States are addressed. The Clients users learn the coordination, agreement and distributed transactions form a core with added topic replication, security Applications and cloud computing have been included to strengthen and augment distributed systems.

Course Outcomes: At the completion of the course the student will be able to

- know the principles and use of remote procedures and methods
- Understand Distributed File Systems their organizations and name services and their use and role in web systems
- Know the Concepts of Timing and Global States and their role on Distributed System
- Understand the Principles of Distributed Transaction and their relevance
- Know the concepts of replication and security
- Know the applications of Distributed Systems, Multi Media Systems
- Understands the concepts of Mobile and Ubiquitous Computing
- Know the principles of Cloud Computing

Course Syllabus:

Unit 1: Basics – Remote method invocation, indirect communication, distributed objects (Ch. 5-8) 5 Periods

Unit 2: Services – P2P Systems, Distributed file Systems, Name Services (Ch. 9-10, 12-13) 8 Periods

Unit 3: Infrastructure -- Time and Global States, Coordination and agreement, Distributed Transactions, Replication, and Security (Ch. 11, 14-18) 12 Periods

Unit 4: Applications -- Mobile and Ubiquitous Computing, Distributed Multimedia Systems, Google (Ch. 19-21) 8 Periods

Unit 5: Cloud Computing -- Introduction to cloud paradigm, Structure of cloud data centers, Computing in the cloud and challenges, MapReduce paradigm (Reference book/papers) 9 Periods

Total: 42 Periods

REFERENCE TEXT:

1. Coulouris, Dollimore, Kindbeg, and Blair, Distributed Systems – Concepts & Design, 5th Edition, Addison-Wesley, 2012 [Chapters 5 –21]

SUGGESTED READING:

1. A Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services, Springer-Verlag, 2012 [Chapters 1-3, 5, for Cloud Computing]

MTCS-202(P) Practicals: DISTRIBUTED SYSTEMS (2 Credits)

Objectives: This course introduces students to the practical aspects of the theoretical concepts techniques and methods learnt.

Course Outcomes: At the completion of the course the student will be able to:

- Do programming that enables one to know the functioning of Internet Protocols, WWW, IPC
- Understand programming with MPI and solve problems
- Gain insight into programming with the help of RPC and RMI
- Learn group communications and processes and threads as visualised in Distributed environment

Suggested Experiments:

* IPC and MPI can be programmed and demonstrated.
* RPC and RMI can be programmed and demonstrated.
* Communications between groups and approaches to shared memory
* Processes and Threads can be programmed and used in a distributed environment
* Learn Hadoop Framework and implement suitable exercises
* Programming atomic commit protocol Locks, Transactions and Time Stamp Ordering through simulations

Reference Text:

1. Coulouris, Dollimore, Kindbeg, and Blair, Distributed Systems – Concepts & Design, 5th Edition, Addison-Wesley, 2012 [Chapters 5 –21]

* * *

MTCS-203 TOPICS IN DATABASE MANAGEMENT SYSTEMS

(3 Credits) (42 Periods)

Pre-requisites:

- A first level course in Database Management Systems

Course Objectives:

- Introduce the internals of a database management system.
- Introduce concurrency and recovery of databases.
- Explain distributed and parallel databases.
- Explain spatial and temporal databases.

Course Outcomes: At the end of the course, the student will be able to understand

- What happens inside a database when a query is submitted?
- The different types of indices and their roles in query processing.
- How relational algebra operations are performed by the database engine.
- How queries can be optimized for quick execution.
- Maintaining ACIDity even in the presence of concurrent transactions.
- How a database can be recovered after a system crash?
- How the relational algebra operations can be parallelized?
- How concurrency of transactions can be achieved in a distributed system.
- What data structures to use for spatial and temporal queries.
- How functional dependencies are addressed in temporal databases.

Course Syllabus:

Unit 1 : DATA STORAGE AND QUERYING 10 periods

Indexing and Hashing, Algorithms for Query Processing and Query Optimization

Unit 2 : TRANSACTION MANAGEMENT 12 periods

Concurrency Control Techniques, Database Recovery Techniques,

Unit 3 : SYSTEM ARCHITECTURE 12 periods

Database-System Architectures, Parallel Databases, Distributed Databases.

Unit 4 : ADVANCED TOPICS 8 periods

Advanced Application Development, Temporal and Spatial Data, Advanced Transaction Processing.

Total 42 periods

Reference Text:

Silberschatz, A., Korth, H. F., and Sudarsham, S. *Database System Concepts*, 6th Edition, McGraw-Hill, (2010)

Chapters 11, 12, 13 (upto 13.4), 15 to 19, 24 to 26

Suggested Readings:

1. Elmasri, R., and Navathe, S. B., Fundamentals of Database Systems, Pearson Education, 4th edition (2007).
2. Ramakrishnan R., and Gherke, *Database Management Systems*, J. McGraw-Hill, (2000) Second Edition.
3. Sunderraman R., *Oracle 10g Programming: A Primer*, Addison-Wesley, (2008)

MTCS-203(P)

Practicals: TOPICS IN DATABASE MANAGEMENT SYSTEMS

(2 Credit)

Course objectives: To provide the student with necessary hands-on skill set for implementing some important concepts in database management system. To make the student practically design and endeavor as a database system.

Course Outcomes: The student will be capable of:

- How to Programme and use indices.
- How to implement some relational algebra operations like join.
- How to Programme some internal processes like lock manager.

Recommended Assignments:

1. Implement B+ trees
2. Nested Loop Join
3. Lock Manager
4. External sorting

Reference Text:

Silberschatz, A., Korth, H. F., and Sudarsham, S., *Database System Concepts*, 6th Edition, McGraw-Hill, (2010)

* * *

STREAM I: INTELLIGENT SYSTEMS AND KNOWLEDGE ENGINEERING

ISKE 1 ARTIFICIAL INTELLIGENCE

(3 Credits) (42 Periods)

Course Objectives :
The Course will introduce students to the mapping of Human Intelligence and Behaviour onto Digital Systems - Computers that Functions as Intelligent Systems and to familiarize students with Concepts, Agents - Modeling, Intelligent Search, Problem Solving, Logic Reasoning, Knowledge Representation, Different Types of Learning.

Course Outcome : At the completion of the Course the student will be able to:
- Know the Theory and Concepts behind Artificial Intelligence.
- Understand Intelligent Agents, Problem Solving, State Space Search, Problem Solving Methods and Search Techniques that requires Intelligence.
- Understand Propositional Logic and First Order Predicate Logic, Applications, Reasoning Methods and Reasoning Process.
- Understand Principles behind Knowledge Representation Techniques, Ontology.
- Understand Concepts behind Learning - Different Types of Learning Models.

Course Syllabus"

Unit 1:
Introduction – what is AI? – Intelligent agents, environments – Solving problems by searching: problem solving agents –Example problems – Uninformed search strategies – Informed search and exploration: Informed search strategies –Heuristic functions– Local search algorithms – Optimization problems. (10 periods)

Unit 2:
Logical Agents: Knowledge Based Agents – Logic – Propositional logic – Reasoning patterns – Propositional Inference – Agents based on propositional logic – First Order Logic :Representation – Using FOL – Knowledge Engineering – Inference in FOL: Unification And Lifting – Forward Chaining – Backward Chaining – Resolution – Examples. (14 periods)

Unit 3:
Knowledge Representation: Ontological Engineering – Categories and objects – Actions situations and events – Mental events and mental objects – Reasoning Systems – Truth maintenance systems (9 periods)

Unit 4:
Learning from Observations: Forms of learning – Inductive learning – Learning Decision trees - Knowledge in Learning – Knowledge in Learning – Explanation based learning – Learning using relevance information (9 periods)
Total (42 Periods)

Reference Text:
1.Stuart J. Russel and Peter, Norvig, Artificial Intelligence – A Modern Approach, Prentice Hall, Pearson Education, 2003.
[Chapters & Sections : 1: 1.1 ; 2: 2.1 to 2.4 ; 3:3.1 to 3.6; 4: 4.1 to 4.3 ; 7: 7.1 to 7.7; 8 : 8.1 to 8.4; 9: 9.1 to 9.5 ; 10: 10.1 to 10.8; 18: 18.1 to 18.3 ; 19: 19.1 to 19.4]

Suggested Readings:
1. George F. Luger and William A. Stubblefield, Artificial Intelligence, Structures and Strategies for Complex Problem Solving, The Benjamin / Cummings Publishing Co, 1993.
2. Artificial Intelligence and Soft Computing, by Amit Konar, CRC Press, 2000.

ISKE 1(P) Practicals: ARTIFICIAL INTELLIGENCE (1 Credit)

Course Objectives :
The Course introduces students the practical aspects, the effects of the theoretical concepts, Techniques and Methods learnt. The students will understand of how the Intelligent Agents, Techniques and Learning Methods that behaves when put into action, and as seen through simulations.

Course Outcome : At the completion of the Course the student will be able to :
- Know the practical aspects of the theory and concepts learnt
- Understand the principles of Propositional and First Order Predicate Logic, Applications reasoning Methods and Reasoning Process through examples and simulations
- Creation of Knowledge Base by implementing Knowledge Representation Techniques such as Semantic Nets, Frames and Ontology Inject skills into a learner.
  Suggested exercises

Syllabus:
- Informed Search Techniques using Heuristics and their Implementation
  i) Steepest Hill Climbing Programming
  ii) A Star Algorithm Programming to find shortest path / optimal path
  iii) Genetic Algorithm Programming and Optimization
- Forward Chaining Programming
- Backward Chaining Programming
- Reasoning by Resolution

Reference Text:

1. Stuart J. Russel and Peter Norvig, Artificial Intelligence – A Modern Approach, Prentice Hall, Pearson Education, 2003.

Suggested Readings:

1. George F. Luger and William A. Stubblefield, Artificial Intelligence, Structures and Strategies for Complex Problem Solving, The Benjamin / Cummings Publishing Co, 1993.

2. Amit Konar, Artificial Intelligence and Soft Computing, CRC Press, 2000.

* * *

ISKE 2 GENETIC ALGORITHMS (3 Credits) (42 Periods)

Course Objectives:

This course will introduce students to the mapping of evolution and evolutionary process namely Chromosomes and their changes through natural evolutionary operators onto a computational paradigm. The representation of chromosomes, objective function, operators and procedures are made known to students to take them forward to computational paradigm to solve the real world problems.

Course Outcome: At the completion of the Course the student will be able to:

- Know the theory and concepts of genetic algorithms
- Understand evolution, evolutionary process, chromosomes and genetic operators
- Understand representations of chromosomes, cross over, mutation and genetic operators.
- Understand the applications and mapping onto computational paradigm and solve real world problems
- Case Studies on the use and applications of genetic operators and genetic algorithms enable the student to get a better knowledge.

Course Syllabus:

Mathematical Foundations for Genetic Algorithms	(8 periods)
Concepts in Genetic Algorithm and their Implementation	(8 periods)
Operators and Techniques in Genetic Search	(8 periods)
Genetic Based Machine Learning	(10 periods)
Applications	(8 periods)
Total	**42 periods**

Reference Text:

1. David E. Goldberg, Genetic Algorithms in Search, Optimization, And Machine Learning, Addison - Wesley Pub. Co., INC. 1989.

ISKE 2(P) Practicals: GENETIC ALGORITHMS (1 Credit)

Course Objectives :

The Course introduces students the practical aspects of genetic algorithms. The representations of the real world data as chromosomes, development of objective functions and applications of genetic operators are the primary aspects in the course,

Course Outcome : At the completion of the Course the student will be able to :

- Know the practical aspects of the theory and concepts of genetic algorithms learnt
- Analyze the data from the real world domain of interest Encoding and representation of data as chromosomes.

Syllabus:

Application of genetic operators such as crossover, mutation etc.
Case Studies : Identifying the domain related real world problems and simulations using genetic algorithms.

Reference Text:

1. David E. Goldberg, Genetic Algorithms in Search, Optimization, And Machine Learning, Addison - Wesley Pub. Co., INC. 1989.

* * *

ISKE 3 NATURAL LANGUAGE PROCESSING (3 Credits) (42 Periods)

Course Objectives:

The Course introduces students to the principles, concepts and theory behind language processing. There is an introduction to languages grammar followed by transformational grammars of natural language. Transition networks and two level processing of moving from grammar to acceptor. The course will enable a student to learn and work on techniques and models to process sentences formed out of a natural language.

Course Outcome: At the completion of the course the student will:

- know the basic concepts and theory of languages and grammar
- the syntax and semantics of the sentences formed out of a natural language is imparted to students
- the basics of grammar and its applications to sentences are highlighted and students are trained with examples
- get thorough treatment on transformational grammars of natural languages that are introduced.
- know Transition Networks: From Grammar to Acceptor are imparted and given training
- know the Two Level Processing Systems namely RTNs and ATNs are introduced with treatment using examples
- get applications of the methods learnt are imparted to students to gain knowledge and confidence

Syllabus:

Introduction to Languages and Grammars - Transformational Grammars of Natural Language - Two-Level Representation - Transition Networks - From Grammar to Acceptor- Two Level Processing Systems RTN's and ATN's - Issues And Applications.

Reference Text:

1. Gilbert K. Krulee, Computer Processing of Natural Language, Prentice Hall 1991.

ISKE 3(P) Practicals: NATURAL LANGUAGE PROCESSING (1 Credit)

Course Objectives :

The Course Introduces Students the practical aspects of Natural Language Processing. The subject encompasses and covers Parsing, Sentence Analysis, Application of Grammar and methods for semantic analysis of sentences, Other Grammars RTNs and ATNs are included for simulations. The course will enable a learner to take up studies in Document Processing

Course Outcome : At the completion of the Course the student will be able to:

- Know the practical aspects of the theory and concepts of Natural Language Processing (NLP).

Recommended Assignments:

- Sentence Parser and application of Grammar through simulations
- Semantic Analysis
- Total Sentence Analysis for document processing
- Transformational Grammar, Transition Networks for simulations
- RTN and ATN networks.

Reference Text:

1. Gilbert K. Krulee, Computer Processing of Natural Language, Prentice Hall 1991.

* * *

ISKE 4 NEURAL NETWORKS

(3 Credits) (42 Periods)

Course Objectives:

The objective of this course is to introduce students to various neural network architectures and the associated learning paradigms such as perceptron's, multilayer perceptron's, radial-basis function networks, support vector machines. Regularization networks, and self-organizing networks.

Students will realize that the connecting thread in all these learning structures is to map the adaptation of various parameters as a non-linear optimization.

Course Outcomes:

At the end of the course students will be able to solve real-world problems such as

- pattern classification as a non-linear feature space partitioning either by estimating the feature space density or by using the feature vectors that lie at the class boundaries.
- input-output functional mapping and then using these as universal approximations for computing outcomes for the inputs that are unseen.
- Topological mapping of input features to codebook vectors through self-organizing maps.

Course Syllabus:

Unit 1: Introduction:

What is Neural Networks? Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs, Network Architectures, Learning Processes: Learning with a Teacher, Learning Without a Teacher: Reinforcement Learning and Unsupervised Learning, Learning Tasks 5 periods

Unit 2: Rosenblatt's Perceptrons:

Introduction, Perceptron, Perceptron Convergence Theorem, The Batch Perceptron Algorithm 3 periods

Unit 3: Multi-Layer Perceptrons:

Preliminaries, Batch Versus On-line Learning, Back-Propagation Algorithm, Summary of BP Algorithm, Heuristics for making BP Algorithm Perform better, Virtues and Limitations of BP Learning, Supervised Learning viewed as an Optimization Problem 4 periods

Unit 4: Radial-Basis Function Networks:

Introduction, Cover's Theorem, Interpolation Problem, Radial Basis Function Networks, K-Means Clustering, Recursive Least-Squares Estimation of the weight vector, Hybrid learning procedure for RBF Networks, Interpretations of Gaussian Hidden Units, Kernel regression and its relation to RBF Networks. 6 periods

Unit 5: Support Vector Machines:
Introduction, Optimal Hyper-plane for Linearly Separable Patterns and Non-separable Patterns, SVM viewed as a Kernel Machine, Design of SVMs, XOR problem 8 periods

Unit 6: Regularization Networks:
Introduction, Hadamard's conditions for well-posedness, Tikhonov's Regularization Theory, Regularization Networks, Generalized Radial-Basis-Function networks 8 periods

Unit 7: Self-Organizing Maps:
Introduction, Two Basic Feature mapping Models, Self-Organizing Maps, Summary of Self-organizing Algorithm, Properties of Feature Map, Contextual Maps Hierarchical Vector Quantization, Kernel Self-Organizing Map. Relationship between Kernel SOM and Kullback-Leibler Divergence 8 periods

Total 42 periods

Reference Text:
Simon Haykin, Neural Networks and Learning Machines: Eastern Economy Edition, Third Edition, 2009.
[Chapters: Introduction (1-6, 8, 9), Chapter 1(1.1-1.4, 1.6, 1.8), Chapter 4(4.1-4.4, 4.6, 4.15, 4.16), Chapter 5(5.1-5.11), Chapter 6(6.1-6.6), Chapter 7(7.1-7.5), Chapter 9(9.1-9.4, 9.6-9.8, 9.10)

Suggested Readings:
Jacek M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company.

ISKE 4(P) Practicals: NEURAL NETWORKS (1 Credit)

Algorithms/Exercises from different units in the syllabus will be implemented in Lab. There are standard tools available in MATLAB, WEKA and other Software tools for solving pattern classification, behavioural prediction, system identification.

Course Outcomes:
Students would have gained experience in mapping the real-world problems to the available architectures and solve the same. In addition, they would be able to choose the right kind of architectures for the chosen problem.

Syllabus:
- Implementation of feed-forward 3 layer neural networks for classification, system identifier, and input-output mapping between two related data sets.
- Implementation of radial-basis function network as universal approximator for prediction.
- Use of SVMs with kernels for solving nonlinear boundary between classes.
- Self-organizing map as a clustering tool.

Reference Text:
Simon Haykin, Neural Networks and Learning Machines: Eastern Economy Edition, Third Edition, 2009.

Suggested Readings:
Jacek M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company.

* * *

ISKE 5 DATA MINING AND DATA WAREHOUSING

(3 Credits) (42 Periods)

Course Objectives :

The Course introduces students to the principles, concepts and theory behind Data Mining and Data Warehousing. Visualization and statistical perspectives pertaining to data mining are given a good treatment followed by Predictive Modeling. Techniques related to data mining, and predictive analytics are imparted. Then Data Warehousing aspects are taken up for the students to gain knowledge in Mining the data. Applications covering Data Mining and warehousing are well studied and presented for the students to gain skills and knowledge

Course Outcome : At the completion of the course the student will :

- know the basic concepts and theory of data mining and data warehousing
- the overview and use of statistics databases machine learning and mining are imparted
- visualization aspects the fundamentals are dealt
- know dimensionality reduction principle and methods are imparted and the student will have a good knowledge
- know the data summarization methods statistical perspectives are dealt.
- know Probabilistic and Deterministic Models are given a good treatment.
- know Clustering Regression Analysis, Time Series Analysis Are Imparted.
- know Predictive Modeling and Methods are imparted to gain a better insight into Mining a Data
- know Pattern Classification Decision Trees are taken up for imparting skills.
- know basic Concepts and Methods in data warehousing to know about Metadata and Mining.

Course Syllabus:

INTRODUCTION

Relation to statistics, databases, machine learning - Taxonomy of data mining tasks - Steps in data mining process - Overview of data mining techniques.

VISUALIZATION AND STATISTICAL PERSPECTIVES

Visualization – Dimension reduction techniques - Data summarization methods - Statistical Perspective - Probabilistic - Deterministic models - Clustering - Regression analysis - Time series analysis - Bayesian learning.

PREDICTIVE MODELING

Predictive Modeling - Classification - Decision trees - Patterns - Association rules - Algorithms.

DATA WAREHOUSING

Design - Dimensional Modeling - Metadata - Performance issues and indexing -VLDB issues – Development life cycle - Merits.

APPLICATIONS

Tools - Applications - Case Studies.

REFERENCE TEXT

1.Usama M. Fayyad, Geogory Piatetsky Shapiro, Padhrai Smyth and Ramasamy Uthurusamy, Advances in Knowledge Discovery and Data Mining, The M.I.T Press, 1996.
2. Jiawei Han, Micheline Kamber, Data Mining Concepts and Techniques, Morgan Kauffmann Publishers, 2000.
3. Ralph Kimball, The Data Warehouse Life Cycle Toolkit, John Wiley & Sons Inc., 1998.
4. Sean Kelly, Data Warehousing in Action, John Wiley & Sons Inc., 1997.
5. A.K. Pujari, Data mining techniques, University press, India, 2001.

ISKE 5(P) Practicals: DATA MINING AND DATA WAREHOUSING (1 Credit)

Course Objectives :

The Course introduces students the practical aspects of Data Mining Methods and Techniques used for Mining the Database. Visualization and Statistical Perspective are given a Practical for students to gain knowledge in fundamentals. Predictive Modeling methods are introduced for the students to get skills in predictive analytics. Practical aspects and working knowledge in data warehousing enables students to gain insight and working with Mining.

Course Outcome : At the completion of the course the student will be able to know the practical aspects of data Mining and data warehousing. The basics of Statistics, databases, and machine learning principles are learnt.

Course Syllabus:

- Dimensionality reduction
- Data Summarisation
- Probabilistic and Deterministic Models
- Clustering Regression analysis and Time Series analysis
- Classification Decision Trees and Association Rule Mining methods

REFERENCE TEXT

1. Usama M. Fayyad, Geogory Piatetsky Shapiro, Padhrai Smyth and Ramasamy Uthurusamy, Advances in Knowledge Discovery and Data Mining, The M.I.T Press, 1996.
2. Jiawei Han, Micheline Kamber, Data Mining Concepts and Techniques, Morgan Kauffmann Publishers, 2000.
3. Ralph Kimball, The Data Warehouse Life Cycle Toolkit, John Wiley & Sons Inc, 1998.
4. Sean Kelly, Data Warehousing in Action, John Wiley & Sons Inc., 1997.
5. A.K. Pujari, Data mining techniques, University press, India, 2001

* * *

ISKE 6 PATTERN RECOGNITION **(3 Credits) (42 Periods)**

Course Objectives:

The objective of this course is to introduce the state-of-the-art various theories that are used in pattern recognition. They primarily depend upon feature space partitioning viz. Bayes theory, linear Classifiers and their limitations, non-linear classifiers, and feature point clustering in n-dimensional space

Course Outcomes:

At the end of the course students will be able to solve real-world pattern recognition and feature space clustering problems using

- Bayes decision theory, Bayes inference, Bayes classifier, and Bayes Networks.
- Linear discriminant functions, logistic discriminant functions, SVM for separable and non-linearly separable classes
- Non-linear classifiers and their combinations
- A host of Clustering algorithms for small and large data set
- In depth of understanding of theory to select the right approach to solve a given problem.

Course Syllabus:

Unit 1: Introduction: (2 periods)

Introduction, Features, Feature Vectors, Classifiers, Supervised, Unsupervised and Semi-Supervised Learning.

Unit 2: Classifiers based on Bayes Theory: (6 periods)

Introduction, Bayes Decision Theory, Discriminant Functions, Bayes Classification for Normal Distributions, Estimation of Unknown Probability Distributions**:** *ML Parameter Estimation, MAP Estimation, Bayesian Inference, Maximum Entropy Estimation, Mixture Models, Non-Parametric Estimation*, the Naïve-Bayes Classifier, the Nearest Neighbor Rule, Bayesian Networks.

Unit 3: Linear Classifiers: (8 periods)

Introduction, Linear Discriminant Functions and Decisions, Hyper-planes, The Perceptron algorithm, Least Square Methods, Mean Square Estimation Revisited, Logistic Discrimination, Support Vector Machines for Separable Classes, SVM for Non-Separable Classes, SVM for Multiclass Case, ϑ-SVM

Unit 4: Nonlinear Classifiers: (8 periods)

XOR Problem, Two Layer Perceptron, Three-Layer Perceptrons, Algorithms based on Exact Classification of Training Set, The Back-Propagation Algorithm, Variation of BP Theme, Choice of Cost Function, Choice of Network Size, Generalized Linear Classifiers, Capacity of d-dimensional space in linear Dichotomies, Polynomial Classifiers, Radial Basis Function Networks, Universal Approximators, Probabilistic Neural Networks, SVM-Nonlinear Case, Beyond SVM Paradigm, Decision Trees, Combining Classifiers, Boosting, Class Imbalance Problem

Unit 5: Clustering: (18 periods)
Introduction, Proximity Measures, Number of Possible Clusterings, Categories of Clustering Algorithms, Sequential Clustering Algorithms, Agglomerative Algorithms, Divisive Algorithms, Hierarchical Algorithms for Large Datasets., Choice of the Best Number of Clusters, Hard Clustering Algorithms, Vector Quantization. Algorithms based on Graph Theory, Competitive Learning algorithms

Total (42 periods)

Reference Text:

1. Sergios Theodoridis and Knostantinos Koutroumbas, Pattern Recognition, Fourth Edition, Elsevier Publications, 2009.
   Chapters: 1, 2, 3, 4, 11, 12.1-12.3, 13, 14.5, 15.1-15.3

Suggested Reading:

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer 2006 First Indian Print 2013.

ISKE 6(P) Practicals: PATTERN RECOGNITION (1 Credit)

Course objectives: To provide the student with necessary skill set for implementing some important learning algorithms on realistic data. To make the student appreciate concepts such as overfitting, training data, test data, model validation, ROC curves etc.

Course outcomes: At the end of this lab course, a student will be able to implement in MATLAB

- Bayesian learning, Bayes and Naive-Bayes Classifier
- SVM classifier with kernel space projections
- Clustering, Hierarchical clustering for large data set.
- Universal approximators

Recommended assignments:

1. Implement learning algorithms for pattern classification of linearly separable and non-separable classes
2. Implement learning algorithms for clustering

Datasets may be taken from standard websites which pertain to realistic scenarios.

Reference Text:

Sergios Theodoridis and Knostantinos Koutroumbas, Pattern Recognition, Fourth Edition, Elsevier Publications, 2009.

* * *

ISKE 7 MACHINE LEARNING

(3 Credits) (42 Periods)

Prerequisite: Introductory Probability and Statistics

Course objectives: To provide students with an in-depth introduction to two main areas of Machine Learning: supervised and unsupervised. The contents will cover some of the main models and algorithms for regression, classification and clustering. Topics will include linear and logistic regression, MLE, probabilistic (Bayesian) inference, clustering and model selection.

Course outcomes:

- Develop an appreciation for what is involved in learning from data.
- Understand some of the important learning algorithms pertaining to classification and regression.
- Appreciate as to how probabilistic framework is applied for building various models of learning
- Understand how to apply a variety of learning models for the tasks of classification, regression and clustering.
- Understand how to perform evaluation of learning algorithms and model selection.

Course Syllabus:

Unit 1: Introduction - Examples of Machine Learning Applications, Classification, Regression, Unsupervised Learning, Reinforcement Learning, Supervised Learning, Vapnik-Chervonenkis (VC) Dimension, Probably Approximately Correct (PAC) Learning, Learning Multiple Classes, Regression, Model Selection and Generalization (4 Periods)

Unit 2: Bayesian Decision Theory – Classification, Losses and Risks, Discriminant Functions, Utility Theory, Association Rules (3 Periods)

Unit 3: Parametric Methods - Maximum Likelihood Estimation, Bernoulli Density, Multinomial Density, Gaussian Density, Bias and Variance, Bayes' Estimator, Parametric Classification, Regression, Model Selection Procedures (3 Periods)

Unit 4: Multivariate Methods - Multivariate Data, Parameter Estimation, Estimation of Missing Values, Multivariate Normal Distribution, Multivariate Classification, Multivariate Regression (5 Periods)

Unit 5: Clustering - Mixture Densities, k-Means Clustering, Expectation-Maximization, Mixtures of Latent Variable Models, Supervised Learning after Clustering, Hierarchical Clustering (5 Periods)

Unit 6: Decision Trees – Univariate Trees, Classification Tree, Regression Trees, Pruning, Rule Extraction from Trees, Learning Rules from Data, Multivariate Trees (3 Periods)

Unit 7: Linear Discrimination - Generalizing the Linear Model, Geometry of the Linear Discriminant, Two Classes Multiple Classes, Gradient Descent, Logistic Discrimination, Discrimination by Regression (4 Periods)

Unit 8: Bayesian Estimation - Estimating the Parameter of a Distribution, Bayesian Estimation of the Parameters of a Function, Use of Basis/Kernel Functions, Bayesian Classification, Gaussian Processes (4 Periods)

Unit 9: Graphical Models - Canonical Cases for Conditional Independence, Example Graphical Models, Naive Bayes' Classifier, Linear Regression, Belief Propagation, Chains, Trees, Poly trees, Junction Trees, Markov Random Fields, Learning the Structure of a Graphical Model, Influence Diagrams (5 Periods)

Unit 10: Design and Analysis of Machine Learning Experiments - Factors, Response, and Strategy of Experimentation, Response Surface Design, Randomization, Replication, and Blocking, Cross-Validation and Resampling Methods, K-Fold Cross-Validation, Bootstrapping, Measuring Classifier Performance, Interval Estimation, Hypothesis Testing, Assessing a Classification Algorithm's Performance, Binomial Test, Approximate Normal Test, t Test, Comparing Two Classification Algorithms, McNemar's Test K-Fold Cross-Validated Paired t Test, Comparing Multiple Algorithms by analysis of variance, Comparison over Multiple Datasets (6 Periods)

Total: 42 Periods

Reference Text:
Introduction to Machine Learning, Ethem ALPAYDIN, The MIT Press, February 2010, ISBN-10: 0-262-01243-X, ISBN-13: 978-0-262-01243-0

Prescribed Chapters: 1, 2, 3, 4, 5, 7, 9, 10, 14, 16, 19

Suggested Readings:
1. Tom Mitchell, Machine Learning, McGraw Hill (Oct 1997).

2. Christopher M. Bishop, Pattern Recognition and Machine Learning, Pub Springer (Aug 2006)
3. Stephen Marsland, Machine Learning: An Algorithmic Perspective, Pub Chapman & Hall/Crc (Apr 2009)

ISKE 7(P) Practicals: MACHINE LEARNING **(1 Credit)**

Course objectives: To provide the student with necessary skill set for implementing some important learning algorithms on realistic data. To make the student appreciate concepts such as overfitting, training data, test data, model validation etc.

Course outcomes: At the end of this lab course, a student will be able to

- Do basic cleaning of the data to suit his/her implementation
- Code the specific algorithm of classification or regression in a language such as python
- Report the performance of the implemented code through necessary graphs/tables
- Analyse the performance and say why or why not the algorithm is behaving in a particular manner

Recommended assignments:

1. Implement learning algorithms for classification
2. Implement learning algorithms for regression
3. Implement learning algorithms for clustering

Datasets may be taken from standard websites which pertain to realistic scenarios.

Reference Text:

Ethem ALPAYDIN, Introduction to Machine Learning, The MIT Press, February 2010, ISBN-10: 0-262-01243-X, ISBN-13: 978-0-262-01243-0

* * *

ISKE 8 Mining of Big Data Sets (2 Credits) (28 Periods)

Course objectives:
The course will discuss data mining and machine learning algorithms for analyzing very large amounts of data. The emphasis will be on Map Reduce and Spark as tools for creating parallel algorithms that can process very large amounts of data.
Topics include: Frequent item sets and Association rules, Near Neighbor Search in High Dimensional Data, Locality Sensitive Hashing (LSH), Dimensionality reduction, Recommendation Systems, Clustering, Link Analysis, Data streams, Mining.

Course outcomes: At the end of the course, a student must be able to

- Understand the significance of scaling up an algorithm for huge datasets
- Implement a few algorithms in Map reduce paradigm
- Understand the importance and implement randomized & hashing algorithms that scale up well with data size without sacrificing much on accuracy

Course Syllabus:

UNIT-I (4 Periods)
Data Mining: Bonferroni's Principle - Hash Functions - Power Laws.
MapReduce and the New Software Stack - Distributed File Systems - Physical Organization of Compute Nodes - Large-Scale File-System Organization – MapReduce - Extensions to MapReduce -Workflow Systems - Recursive Extensions to MapReduce - Pregel

The Communication Cost Model- Communication-Cost for Task Networks - Complexity Theory for MapReduce - Reducer Size and Replication Rate - A Graph Model for MapReduce Problems - Case Study: Matrix Multiplication

UNIT-II (6 Periods)
Finding Similar Items - Applications of Near-Neighbor Search - Jaccard Similarity of Sets-Collaborative Filtering as a Similar-Sets Problem - Shingling of Documents - k-Shingles - Choosing the Shingle - Hashing Shingles - Shingles Built from Words - Similarity-Preserving Summaries of Sets - Matrix Representation of Sets - Minhashing - Locality-Sensitive Hashing for Documents - LSH for Minhash Signatures - Analysis of the Banding Technique - Combining the Techniques - Distance Measures - The Theory of Locality-Sensitive Functions - Applications of Locality-Sensitive Hashing - Methods for High Degrees of Similarity - Finding Identical Items

UNIT-III (6 Periods)
Mining Data Streams - The Stream Data - Sampling Data in a Stream - A Motivating Example - Filtering Streams - The Bloom Filter - Counting Distinct Elements in a Stream - Estimating Moments - Counting Ones in a Window - Query Answering in the DGIM Algorithm – Decaying Windows

Link Analysis - PageRank - Early Search Engines and Term Spam - Efficient Computation of PageRank - PageRank Iteration Using MapReduce - Other Efficient Approaches to PageRank Iteration - Topic-Sensitive PageRank - Link Spam - Architecture of a Spam Farm - Analysis of a Spam Farm - Combating Link Spam – Trust Rank - Spam Mass

UNIT-IV (6 Periods)
Frequent Itemsets - Association Rules - Finding Association Rules with High Confidence - Use of Main Memory for Itemset Counting - Monotonicity of Itemsets - Handling Larger Datasets in Main Memory - The Multistage Algorithm - The Multihash - The SON Algorithm and MapReduce - Counting Frequent Items in a Stream

UNIT- V (6 Periods)
Clustering - Introduction to Clustering Techniques - Clustering Strategies - The Curse of Dimensionality - Hierarchical Clustering - Hierarchical Clustering in Non-Euclidean Spaces - K-means Algorithms - The Algorithm of Bradley, Fayyad, and Reina - The CURE Algorithm - Clustering for Streams and Parallelism

Reference Text: Mining of Massive Datasets by Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman Cambridge University Press, 2nd Edition

The text book will be supplemented with research papers and assignments & Projects designed for the course by the instructor.

ISKE 8(P) Practicals: Mining of Big Data Sets **(2 Credits)**

Course objectives: To provide the student with necessary skill set for implementing some important learning algorithms on large data. To make the student appreciate concepts such as scaling in computation with data size, randomization and hashing strategies for getting required performance and accuracies etc.

Course outcomes: At the end of this lab course, a student will be able to

- Model some big data related algorithms for Map-Reduce paradigm
- Implement specific algorithms on Map-Reduce platforms such as SPARK
- Study the performance of the implemented code through experimentation on standard datasets
- Analyse about the performance and say why or why not the algorithm is behaving in a particular manner

Course Syllabus:
Map Reduce, Finding Similar Items, Mining Data Streams, Link Analysis,
Frequent Item-sets, Clustering, Recommendation System related Algorithms/Exercises from the syllabus will be implemented in Lab. Students may be encouraged to do a mini project.

Reference Text: Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, Mining of Massive Datasets, Cambridge University Press, 2nd Edition.

ISKE 9 Deep Learning **(2 Credits) (28 periods)**

Course Objectives:

- To Introduce deep learning (DL) algorithms including convolutional neural networks (CNN), recurrent neural networks (RNN) and its variants viz. LSTM and GRU
- To train on how to fine tune hyper parameters of DL algorithms
- To impart concepts that help identify suitable applications for CNN, RNN, LSTM and GRU and study them

Course Outcomes: A student at the end of course should be able to

- Decide if DL is suitable for a given problem
- Choose appropriate DL algorithm to solve the problem with appropriate hyper parameter setting
- Feel comfortable to read and understand DL articles from reputed conferences, journals including NIPS, CVPR, ICCV, ICML, PAMI etc.

Course Syllabus:

Unit I : Introduction - What is Deep Learning? – Perceptron and Multi-layer Perceptron – Hebbian Learning - Neural net as an Approximator - Training a neural network - Perceptron learning rule - Empirical Risk Minimization - Optimization by gradient descent (4 Periods)

Unit 2: Back Propagation - Calculus of Back Propagation (4 Periods)

Unit 3: Convergence in Neural networks - Rates of Convergence – Loss Surfaces – Learning rate and Data normalization – RMSProp, Adagrad and Momentum (4 Periods)

Unit 4: Stochatic Gradient Descent - Acceleration – Overfitting and Regularization – Choosing a Divergence Loss Function – Dropout – Batch Normalization (4 Periods)

Unit 5: Convolutional Neural Networks (CNN) - Weights as Templates – Translation Invariance – Training with shared parameters – Arriving at the convolutional model - Mathematical details of CNN – Alexnet – Inception – VGG - Transfer Learning (6 Periods)

Unit 6: Recurrent Neural Networks (RNNs) - Modeling sequences - Back propagation through time - Bidirectional RNNs - Exploding/vanishing gradients - Long Short-Term Memory Units (LSTMs) (6 Periods)

Total 28 Periods

Reference Text:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, Online book, 2017
2. Michael Nielsen, Neural Networks and Deep Learning, Online book, 2016

Suggested Readings:

1. http://cs231n.github.io/
2. https://www.coursera.org/specializations/deep-learning
3. https://cilvr.cs.nyu.edu/lib/exe/fetch.php?media=deeplearning:2015:a4.pdf
4. http://deeplearning.cs.cmu.edu/

ISKE 9(P) Practicals: Deep Learning **(2 Credits) (28 periods)**

Course Objectives:

- To Introduce PyTorch/TensorFlow as a deep learning framework
- Train on Implementing various DL algorithms studied in theory
- To become good in analyzing the influence of hyper parameters
- To have hands on through a mini project

Course Outcomes: A student should be

- Able to understand and analyse existing codes in PyTorch/TensorFlow from Github
- Able to implement proposed algorithms on reputed frameworks or design a his/her own architecture to suite the need

Lab exercises can be based on the recommended list given below. Lab exercises could be implemented in Python 3 and Tensor Flow/PyTorch.

Lab 1: Implement a shallow network in Python for a binary classification problem

Lab 2: Make the network of Lab1 deeper and compare the performance of two networks on various aspects

Lab 3: Build a fully connected deep network to classify Cifar10 dataset

Lab 4: Analyse convergence of Lab3 network and improvise using various methods studied to accelerate convergence

Lab 5: Analyse over-fittingness of lab3 network and improvise using regularization methods studied

Lab 6: Build a cnn using basic python and numpy to classify Cifar10 data

Lab 7: Play with Pytorch / Tensorflow by going through online tutorials

Lab 8: Implement lab 7 network in Pytorch/Tensorflow

Lab 9: Train a RNN language model to do word-level and character level prediction

Reference Text:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, Online book, 2017
2. Michael Nielsen, Neural Networks and Deep Learning, Online book, 2016

Suggested Readings:

1. http://cs231n.github.io/
2. https://www.coursera.org/specializations/deep-learning
3. https://cilvr.cs.nyu.edu/lib/exe/fetch.php?media=deeplearning:2015:a4.pdf
4. http://deeplearning.cs.cmu.edu/

STREAM II: ADVANCED COMPUTER NETWORKS

ACN 1 TELECOM NETWORKING

(3 Credits) (42 Periods)

Course objectives: This course provides an introduction to the networking principles & techniques of design, implementation, and analysis of telecommunications networks which are instrumental technologies underlying many modern systems. Topics include: basics of transmission and switching, network topologies, architecture, switching techniques, network design, protocols, Local Area Network (LAN), Wide Area Network (WAN), CCITT Signaling System No. 7, Cellular and PCS Radio Systems, SONET/SDH, Asynchronous Transfer Mode (ATM)

Course outcomes: At the end of the course the student will be able to:

- Describe various network services, protocols and architectures.
- Analyze various routing algorithms/protocols.
- Appreciate the fundamental difference between LAN and WAN protocols and understand their architecture
- Analyze resource allocation and congestion control methods.
- Understand and appreciate principles of error management, performance characteristics and QoS
- Understand and explain some of the modern technologies such as cellular and PCS Radio Systems, ATM etc.

Course Syllabus:

Unit 1: (8 Periods)
Introductory Concepts of Telecommunications; Transmission and Switching;
(Chapters 1 to 4 from key text)

Unit 2: (10 Periods)
Digital Networks; Signaling; Local and long-distance networks.
(Chapters 6 to 10 from key text)

Unit 3: (15 Periods)
Enterprise Networks: Wide Area Networks; CCIITT Signaling System No. 7.
(Chapters 11 to 14 from key text)

Unit 4: (9 Periods)
Cellular and PCS Radio Systems, Advanced Broadband Digital
Transport Formats: SONET/SDH; Asynchronous Transfer Mode (ATM).
(Chapters 18 to 20 from key text)

Total 42 Periods

Reference Text

1. Roger L. Freeman, Fundamentals of Telecommunications, 2nd Edition, Wiley-IEEE Press; 2 Edn., 2005, ISBN-10: 0471710458 ISBN-13: 978-0471710455.

Suggested Readings

2. Roger L. Freeman, Telecommunication System Engineering, Wiley-Interscience; 4th edition, 2004, ISBN-10: 0471451339 ISBN-13: 978-0471451334.

3. Thomas Plevyak, Veli Sahin, Next Generation Telecommunications Networks, Services, and Management, Wiley-IEEE Press, 2010.

ACN 1(P) Practicals: TELECOM NETWORKING (1 Credit)

Course objectives:
This course provides practical training to supplement the networking principles & techniques of design, implementation, and analysis imparted in the theory course.

Course outcomes: At the end of the course the student will have the practical knowledge of some of the theoretical concepts taught in the theory course.

Syllabus: Some simulation studies based on concepts/technology topics from the theory course can be undertaken on standard simulators such as OMNET++, NS(3) or Wireshark.

Reference Text

1. Roger L. Freeman, Fundamentals of Telecommunications, 2nd Edition, Wiley-IEEE Press; 2nd Edn., 2005, ISBN-10: 0471710458 ISBN-13: 978-0471710455.

2. http://ns3simulation.com/list-of-network-simulators/

* * *

ACN 2 NETWORK SECURITY **(3 Credits) (42 Periods)**

Course objectives:

The objectives of the course are to train students who will understand the mathematical principles of cryptography mechanisms and standards, and who will be able to understand how these mechanisms are used in designing secure network protocols. The course will also train students on understanding common and recent network security threats and mechanisms to handle such threats.

Course outcomes: At the end of the course, the student will be able to:

- Understand the mathematical foundations of standard cryptography algorithms and mechanisms including DES, RSA, DSS and other standards used to secure network data transmitted from source to destination
- Understand how these algorithms and mechanisms are applied to different network protocol layers, and in particular understand the internal operations of protocols such as IPSec, TLS, Digital Signature, etc.
- Understand common user authentication protocols and access control mechanisms
- Understand the different passive and active network attack techniques
- Understand the concepts of firewalls, packet filters and related mechanisms to deal with network attacks
- Design a system infrastructure for any new network system that requires to be secured from attacks

Course Syllabus:

Unit 1: (3 Periods)

Introduction-Motivating examples, Basic concepts-confidentiality, integrity, availability, security policies, security mechanisms, assurance, Basic Cryptography-Historical background, Transposition /Substitution, Caesar Cipher, Introduction to Symmetric crypto primitives, Asymmetric crypto-primitives, and Hash functions.

Unit 2: (3 Periods)

Secret Key Cryptography-Data Encryption Standard (DES), Encrypting large messages (ECB, CBC, OFB, CFB, CTR), Multiple Encryption DES (EDE).

Unit 3: (3 Periods)

Message Digests, Strong and weak collision resistance, The Birthday Paradox
MD5, SHA-1.

Unit 4: (6 Periods)

Public Key Cryptography-Applications, Theory: Euclidean algorithm, Euler Theorem, Fermat Theorem, Totient functions, multiplicative and additive inverse RSA, Selection of public and private keys.

Unit 5: (6 Periods)
Authentication-Security Handshake pitfalls, Online vs. offline password guessing, Reflection attacks, Per-session keys and authentication tickets, Key distribution centers and certificate authorities

Unit 6: (6 Periods)
Trusted Intermediaries-Public Key infrastructures, Certification authorities and key distribution centers, Kerberos

Unit 7: (5 Periods)
Real-time Communication Security Introduction to TCP/IP protocol stack, Implementation layers for security protocols and implications, IPsec: AH and ESP, IPsec: IKE, SSL/TLS

Unit 8: (5 Periods)
Electronic Mail Security, Distribution lists, Establishing keys, Privacy, source authentication, message integrity, non-repudiation, proof of submission, proof of delivery, message flow confidentiality, anonymity, Pretty Good Privacy (PGP)

Unit 9: (5 Periods)
Firewalls and Web Security, Packet filters, Application level gateways, Encrypted tunnels, Cookies, Web security problem.

Total 42 Periods

Reference Text:

1. Charlie Kaufman, Radia Perlman and Mike Speciner, Network Security: Private Communication in a public world, IInd Edn, ISBN 0-13-046019-PrenticeHall PTR, 2002.

Suggested Readings:

1. William Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall, 4th Edn, 2009.

2. William Cheswick, Steven M. Bellovin and Aviel D. Rubin, Firewalls and Internet Security: Repelling the Wily Hacker, 2nd edition, Addison- Wesley Profession, 2000.

ACN 2(P) Practicals: NETWORK SECURITY **(1 Credit)**

Course objectives: The objectives of the course are to train students who will understand the practical aspects of cryptography mechanisms and standards

Course outcomes: At the end of the course, the student will be able to Understand the mathematical foundations of standard cryptography algorithms and implement the mechanisms including DES, RSA, DSS and other standards used to secure network data transmitted from source to destination.

Students will implement:

- Symmetric Key based DES, 3DES, AES, and Public/Private Key RSA Cryptographic algorithms
- Message Authentication via Hash functions MD5, SHA-1.

Reference Text:

1. Charlie Kaufman, Radia Perlman and Mike Speciner, Network Security: Private Communication in a public world, 2nd Edn, ISBN 0-13-046019-PrenticeHall PTR, 2002

* * *

ACN 3 WIRELESS AND MOBILE NETWORKS

(3 Credits) (42 Periods)

Course objectives: The objectives are to impart fundamental knowledge of wireless network architectures, algorithms, protocols and applications, with special emphasis on local area networks, wide area networks, mobile ad hoc networks and wireless sensor networks.

Course outcomes: Upon completing the course, the student will be able to:
- Understand how wireless local area networks operate and understand the corresponding design principles and protocols
- Understand the protocols and operation of wireless wide area networks, such as cellular networks
- Understand the challenges, protocols and mechanisms used for supporting medium access, routing and quality of service in mobile ad hoc networks and wireless sensor networks.

Course Syllabus:

Unit 1: (6 periods)
Wireless Local Area Networks - IEEE 802.11 family and related protocols; (Chapters 1,2 from key text)

Unit 2: (12 periods)
Wireless Wide Area Networks - 3G/LTE/WiMAX;
(Chapters 3 from key text, Chapters 1 and 2 from Ref Book 4by Martin Sauter)

Unit 3: (15 periods)
Mobile Ad Hoc Networks - Medium access control and Routing protocols;
(Chapters 5 to 7 from key text)

Unit 4: (9 periods)
Quality of service in Mobile Ad Hoc Networks, Wireless Sensor Networks - IEEE 802.15.4.
(Chapters 10.1-10.5, 12 from key text)

Total 42 Periods

Reference Text Book:

1. C. Siva Ram Murthy and B. S. Manoj, Ad Hoc wireless networks: architectures and protocols, Prentice Hall PTR, 2004.

Suggested Readings:

2. Jochen H. Schiller, Mobile Communications, 2nd edition, Addison-Wesley, 2003, ISBN 0-321-12381-6.

3. Wireless Communications & Networks (2nd Edition), William Stallings, 2004.

4. Martin Sauter, Beyond 3G - Bringing Networks, Terminals and the Web Together: LTE, WiMAX, IMS, 4G Devices and the Mobile Web 2.0 Wiley; 1 edition (February 17, 2009) ISBN-10: 0470751886.

5. Kaveh Pahlavan, Principles of Wireless Networks: A Unified Approach, 2nd Edition, 2012 (Expected), ISBN-13: 978-0470697085, Wiley.

6. Erik Dahlman, Stefan Parkvall, Johan Skold, 4G: LTE/LTE-Advanced for Mobile Broadband, ISBN-10: 012385489X, ISBN-13: 978-0123854896, Academic Press, 2011.

7. Jonathan Loo, Jaime Lloret Mauri, Jesus Hamilton Ortiz, Mobile Ad Hoc Networks: Current Status and Future Trends, CRC Press, 2011, Edited Book.

8. Shih-Lin Wu, Yu-Chee Tseng, Wireless Ad Hoc Networking: Personal-Area, Local-Area, and the Sensory-Area Networks, Auerbach Publications, 2007, Edited Book.

9. Rajeev Shorey, A. Ananda, Mun Choon Chan, Wei Tsang Ooi, Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions, Wiley-IEEE Press, 2006, ISBN-13: 978-0471718161, Edited Book.

ACN 3(P) Practicals: WIRELESS AND MOBILE NETWORKS

(1 Credit)

Course objectives: This course provides practical training to supplement the objectives theory course

Course outcomes: At the end of the course the student will have the practical knowledge of some of the theoretical concepts taught in the theory course.

Syllabus: Some simulation studies based on concepts/technology topics from the theory course can be undertaken on standard simulators such as OMNET++, NS(3) or Wireshark.

Reference Text Book:

1. C. Siva Ram Murthy and B. S. Manoj, Ad Hoc wireless networks: architectures and protocols, Prentice Hall PTR, 2004.
2. http://ns3simulation.com/list-of-network-simulators/

***

ACN 4 ADVANCED COMPUTER NETWORKS

(3 Credits) (42 Periods)

Course objectives:
The objectives of the course include training students in the architectures, protocols, and algorithms used in the design and implementation of computer network routers.

Course outcomes: Upon completing the course, the students will be able to:

- Understand the internal architectural choices of network routers
- Understand how IP packets are and classified, forwarded in a router
- Understand the scheduling and routing/switching mechanisms and algorithms used to support quality of service in networks

Course Syllabus:

Unit 1: Overview of prerequisites (8 periods)
Introduction to Networking and Network Routing.
(Chapters 1 to 5, 7 from key text)

Unit 2: Router Architectures (10 periods)
Router Architectures IP Address Lookup Algorithms, IP Packet Filtering and Classification.
(Chapters 14 to 16 from key text)

Unit 3: Quality of Service Routing (12 periods)
Quality of Service Routing, MPLS and GMPLS, Routing and Traffic Engineering with MPLS, VoIP Routing. (Chapters 17 to 20 from key text)

Unit 4: Packet queuing and scheduling (12 periods)
Switching Packets, Packet Queuing and Scheduling, Traffic Conditioning.
(Chapters 21 to 23)

Total: 42 periods

Reference Textbook(s):

1. Medhi and Ramaswami, Morgan-Kaufmann, Network Routing: Algorithms, Protocols and Architectures, 2007, ISBN 13: 978-0-12-088588-6, ISBN 10:0-12-088588-3.

Suggested Readings:

2. George Varghese, Network Algorithmic: An Interdisciplinary Approach to Designing Fast Networked Devices, Morgan-Kaufmann, 2005.
3. Michal PioÌ ro and Deepankar Medih, Routing, Flow, and Capacity Design in Communication and Computer Networks, Morgan-Kaufmann, 2004.
4. James D. McCabe, Network Architecture, Analysis, and Design, Morgan-Kaufmann, 2007.

ACN 4(P) PRACTICALS: ADVANCED COMPUTER NETWORKS

(1 Credit)

Course objectives:

The objectives of this practical course is to reinforce the concepts of the theory course through practical exercises in a computer lab. The concepts pertain to the architectures, protocols, and algorithms used in the design and implementation of computer network routers.

Course outcomes:

At the end of the course the student will have the practical knowledge of some of the theoretical concepts taught in the theory course

Recommended exercises: Simulation excises can be undertaken for the following topics

- IP packets capture and study the classification and forwarding in a router
- Scheduling and routing/switching mechanisms and algorithms used to support quality of service in networks

Reference Textbook(s):

1. Medhi and Ramaswami, Network Routing: Algorithms, Protocols and Architectures, Morgan-Kaufmann, 2007, ISBN 13: 978-0-12-088588-6, ISBN 10:0-12-088588-3.

2. http://ns3simulation.com/list-of-network-simulators/

* * *

STREAM III: HUMAN COMPUTER INTERACTION

HCI 1 DIGITAL IMAGE PROCESSING (3 Credits) (42 Periods)

Course Objectives:
This course will provide an introduction to the fundamental concepts of digital image processing. The course will focus on image fundamentals, enhancement, restoration, frequency domain filtering and segmentation. Students will be introduced to usage of mathematics for image processing procedures.

Course Outcomes: Upon completion of the course, the students will be

- Having a broad understanding of fundamentals of digital image processing in use and will be able to analyze the components of a digital image processing system.
- Able to apply basic image enhancement techniques.
- Familiar with the modeling and analysis of image restoration in presence of noise and degradation.
- Able to understand basic filters and the essential concepts in frequency domain filtering.
- Understand and interpret basic image segmentation techniques and their application scenarios.

Course Syllabus:

Unit 1: Introduction to Digital Image Processing- Fundamental Steps in Digital Image Processing - Components of an Image Processing System (4 periods)

Unit 2: Digital Image Fundamentals – Elements of visual perception – Light and electromagnetic spectrum – image sensing and acquisition - Image Sampling and Quantization- Basic Relationships between Pixels – An introduction to mathematical tools used in digital image processing (4 Periods)

Unit 3: Intensity Transformations and Spatial Filtering – Some basic intensity transformation functions - Histogram Processing – Fundamentals of spatial filtering- Smoothing and sharpening spatial filters - combining spatial enhancement methods (8 Periods)

Unit 4:Filtering in the frequency domain – Sampling and the Fourier transform of sampled functions – basics of filtering in the frequency domain – image smoothing and sharpening using frequency domain filters – selective filtering (8 Periods)

Unit 5: Image Restoration and Reconstruction – Model for image degradation and restoration process – noise models – restoration in the presence of noise only spatial filtering – periodic noise reduction by frequency domain filtering – linear position invariant degradations – estimating the degradation function – inverse filtering (10 periods)

Unit 6: Image Segmentation- Point, line and edge detection – thresholding – region based segmentation (8 Periods)

Total: 42 Periods

Reference Textbook(s):

1. Rafael. C. Gonzalez & Richard E. Woods, Digital Image Processing, 3rd Edition, Pearson Education, 2002. [Chapters 1, 2, 3.1 to 3.7, 4.1 to 4.10, 5.1 to 5.7, 10.1 to 10.4]
2. Relevant research papers selected for the course by the instructor

Suggested Readings:

1. Maria Petrou and Costas Petrou, Image Processing – The Fundamentals, Second Edition, John Wiley and Sons, 2010

2. Anil. K. Jain, Fundamentals of Digital Image Processing, Eastern Economy Edition, Prentice Hall of India 1997

HCI 1(P) Practicals: DIGITAL IMAGE PROCESSING (1 Credit)

Course Objectives: In this course, the student will gain hands-on training on various themes discussed in the (HCI-1) digital image processing lectures. This course will clarify the concepts and principles and help enhance the theory-to-code skill.

Course Outcomes: By the end of this course, the students will be introduced to the basic know-how to convert relevant image processing mathematics to code.

Syllabus:

- Read, write and manipulate digital images.
- write a Programme to implement basic image processing algorithms related to enhancement, restoration, filtering and segmentation.

Reference Textbook(s):

Rafael. C. Gonzalez & Richard E. Woods, Digital Image Processing, 3rd Edition, Pearson Education, 2002.

* * *

HCI 2 MEDICAL IMAGE PROCESSING (3 Credits) (42 Periods)

Course Objectives:
This course will provide an introduction to various imaging modalities such as CT, Ultrasound, X-Ray, MRI in the medical imaging field. The mathematics behind the subject (transform techniques) are also covered for providing an in depth understanding of the imaging techniques. The course also achieves a node acquaintance of various methods and aspects of medical imaging domain.

Course Outcomes: Upon completion of the course, the students will be

- Having a broad understanding of fundamentals of medical imaging modalities such as X-Ray, CT, Ultrasound and MRI.
- Able to apply basic mathematical tools such as transform techniques on the images
- Familiar with the modeling and analysis of reconstruction, relaxation and contrast enhancement mechanisms.
- Visualize medical data in an appropriate software

Course Syllabus:
Historical perspective -Generic Principles – modality – contrast – SNR – resolution – toxicity - Measurements and Modeling: Review of Linear Systems and Models – Basic Model for Tomography - Sampling - Fourier and Hankel transforms - k-space.

XRay projection radiography – Reconstruction in X-Ray Tomography - Computerized Tomography - acquisition and reconstruction methods - relaxation and contrast mechanisms – applications - Nuclear medicine - radio nuclides, PET, SPECT imaging – Applications of Probability : PET.

Ultrasound Imaging - echo equation - beam forming - Medical Image Processing - physics of Magnetic resonance imaging - MRI reconstruction, functional MRI.
Fuzzy and Neuro Fuzzy Systems: Medical Image Analysis and Processing – Wavelets and Fuzzy gated SPECT Images of Ventricles.

Visualization of medical imaging data-segmentation applications.

Reference Books:

1. Albert Macovski, Medical Imaging Systems, Prentice Hall, 1983.

2. Joseph Hornak, The Basics of MRI, Online at http://www.cis.rit.edu/htbooks/mri

3. Charles L. Epstein, Introduction to Mathematics of Medical Imaging, Pearson Education, Prentice Hall, NJ, 2003.

4. H.N. Teodorescu, L.C. Jain, Abraham Kandel, Fuzzy and Neuro Fuzzy Systems in Medicine, Computational Intelligence, CRC Press, 1999.

Suggested Reading:

1. John L Semmlow, Biosignal and Biomedical Image Processing: MATLAB Based Applications, CRC Press.

2. Kavyan Najarian, Biomedical Signal and Image Processing, CRC Press.

3. Isaac Bankmem, Handbook of Medical Imaging: Processing and Analysis, Academic Press, 2000.

4. Anil. K. Jain, Fundamentals of Digital Image Processing, Eastern economy ed., Prentice Hall of India, 1997.

HCI 2(P) Practicals: MEDICAL IMAGE PROCESSING

(1 Credit)

Course Objectives: This course familizarizes the student with various phase of handling medical images - image acquisition, preprocessing and enhancement, processing, segmentation and analysis. The software used is 3D Slicer.

Course Outcomes: Upon completion of the lab course, the students will be

- able to acquire medical images in 3D slicer and learn various aspects of manipulating the image
- preprocessing and processing the image
- visualize medical data in an appropriate software

Reference Books:

1. Albert Macovski, Medical Imaging Systems, Prentice Hall, 1983.

2. Joseph Hornak, The Basics of MRI, Online at http://www.cis.rit.edu/htbooks/mri

3. Charles L. Epstein, Introduction to Mathematics of Medical Imaging, Pearson Education, Prentice Hall, NJ, 2003.

4. H.N. Teodorescu, L.C. Jain, Abraham Kandel, Fuzzy and Neuro Fuzzy Systems in Medicine, Computational Intelligence, CRC Press, 1999.

* * *

HCI 3 COMPUTER VISION

(3 Credits) (42 Periods)

Prerequisite: Basic course on Image processing.

Course Objectives:

This course will provide an extension to the material provided in (HCI-I) digital image processing course. Students will be provided an overview of relevant mathematics and data structures commonly used in image analysis algorithms. Students will also be made familiar with basics of representation, recognition and image understanding.

Course Outcomes: Upon completion of the course, the students will be

- Having an understanding of commonly prevalent computer vision techniques with an awareness of research challenges available therein.
- Able to attempt solutions to current computer vision problems.

Course Syllabus:

Unit 1: Review of pre-requisites-I (2 periods)

Motivation, Image Representation and Image Analysis Tasks, Image Representations, a Few Concepts - Image Digitization, Sampling, Quantization, Digital Image Properties, Metric and Topological Properties of Digital Images, Histograms, Entropy, Image Quality, Noise in Images, Color Images, Color Spaces, Cameras: An Overview

Unit 2: Review of pre-requisites-II

The Image, its Mathematical and Physical Background (3 periods)

Overview / Linearity / The Dirac Distribution and Convolution / Linear Integral Transforms / Images as Linear Systems / Introduction to Linear Integral Transforms / 1D Fourier Transform / 2D Fourier Transform / Sampling and the Shannon Constraint / Discrete Cosine Transform / Wavelet Transform / Eigen-Analysis / Singular Value Decomposition / Principle Component Analysis / Other Orthogonal Image Transforms / Images as Stochastic Processes / Images as Radiometric Measurements / Image Capture and Geometric Optics / Lens Aberrations and Radial Distortion / Image Capture from a Radiometric Point of View / Surface Reflectance /

Unit 3: Data Structures for Image Analysis (2 periods)

Levels of Image Data Representation / Traditional Image Data Structures / Matrices / Chains / Topological Data Structures / Relational Structures / Hierarchical Data Structures / Pyramids / Quadtrees / Other Pyramidal Structures

Unit 4: Segmentation (7 periods)

Watershed Segmentation / Region Growing Post-Processing / Matching / Matching Criteria / Control Strategies of Matching / Evaluation Issues in Segmentation / Supervised Evaluation / Unsupervised Evaluation/Mean Shift Segmentation / Active Contour Models - Snakes /

Traditional Snakes and Balloons / Extensions / Gradient Vector Flow Snakes / Geometric Deformable Models - Level Sets and Geodesic Active Contours / Towards 3D Graph-Based Image Segmentation / Simultaneous Detection of Border Pairs / Sub-optimal Surface Detection / Graph Cut Segmentation / Optimal Single and Multiple Surface Segmentation

Unit 5: Shape Representation and Description (8 periods)
Region Identification / Contour-Based Shape Representation and Description / Chain Codes / Simple Geometric Border Representation / Fourier Transforms of Boundaries / Boundary Description using Segment Sequences / B-Spline Representation / Other Contour-Based Shape Description Approaches / Shape Invariants / Region-Based Shape Representation and Description / Simple Scalar Region Descriptors / Moments / Convex Hull / Graph Representation Based on Region Skeleton / Region Decomposition / Region Neighborhood Graphs / Shape Classes

Unit 6: Object Recognition (10 periods)
Knowledge Representation / Statistical Pattern Recognition / Classification Principles / Classifier Setting / Classifier Learning / Support Vector Machines / Cluster Analysis / Neural Nets / Feed-Forward Networks / Unsupervised Learning / Hopefield Neural Nets / Syntactic Pattern Recognition / Grammars and Languages / Syntactic Analysis, Syntactic Classifier / Syntactic Classifier Learning, Grammar Inference / Recognition as Graph Matching / Isomorphism of Graphs and Sub-Graphs / Similarity of Graphs / Optimization Techniques in Recognition.

Unit 7: Image Understanding (10 Periods)
Image Understanding Control Strategies / Parallel and Serial Processing Control / Hierarchical Control / Bottom-Up Control / Model-Based Control / Combined Control / Non-Hierarchical Control / RANSAC: Fitting via Random Sample Consensus / Point Distribution Models / Active Appearance Models / Pattern Recognition Methods in Image Understanding / Classification-Based Segmentation / Contextual Image Classification / Boosted Cascade of Classifiers for Rapid Object Detection / Scene Labeling and Constraint Propagation / Discrete Relaxation / Probabilistic Relaxation / Searching Interpretation Trees / Semantic Image Segmentation and Understanding / Semantic Region Growing

Total: 42 Periods

Reference Text

1. Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis, and Machine Vision, 3rd Ed, March 19, 2007, Thomson Brooks/Cole Pub.

Coverage of Reference Text :
Relevant sections from Chapters 1 to 10 from Key Text.

Suggested Readings: Latest editions of

1) David A. Forsyth and Jean Ponce, **"**Computer Vision: A Modern Approach" Prentice Hall of India, 2006 Emanuele Trucco, Alessandro Verri, "Introductory Techniques for 3-D Computer Vision", Prentice Hall, 1998.
2) Robert M. Haralick and Linda G. Shapiro, "Computer and Robot Vision", Addison-Wesley.
3) Mubarak Shah, "Fundamentals of Computer Vision", Free E-Book available at Authors site: http://vision.eecs.ucf.edu/faculty/shah.html

HCI 3(P) PRACTICALS: COMPUTER VISION (1 Credit)

Course Objectives:
In this course, the student will implement and analyze chosen algorithms from computer vision literature to understand in depth the lectures of (HCI-3) Computer Vision course. The student will also analyze simple real life applications and their implementations demonstrating computer vision challenges.

Course Outcomes: By the end of this course, the students will be

- Able to prototype image analysis and understand solutions for specific scenarios.

Syllabus:
Algorithms/Exercises from different units in the theory course can be taken up for Lab exercise.

* * *

HCI 4 ADVANCED TOPICS IN IMAGE PROCESSING

(3 Credits) (42 Periods)

Course Objectives:
The course familiarizes the students with an advanced treatment of the various tools of the area of Image Processing. After a brief review of mathematical topics, the main image processing concepts such as image restoration, segmentation and classification are discussed. A special emphasis is given to, functional analysis, partial differential equations and calculus of variations approach.

Course Outcomes: Upon completion of the course the student will

- be Familiar with the Functional Analytic methods in Image Processing
- be able to make a variational formulation (wherever possible) of a task in image processing
- learn some of the modern algorithms for restoration, segmentation and classification of images.
- prepare for research skill associated with the domain of Image Processing
- know how to make a basic implementation for solving the PDEs that emerge from the formulation

Course Syllabus:

Unit 1 : Mathematical Preliminaries (4 Periods)
Direct methods in the Calculus of Variations, The Space of Bounded Variation, Viscosity Solutions in PDEs, Curvature, Dominated Convergence Theorem.

Unit 2 : Image Restoration (8 Periods)
Image Degrading, The Energy Method, PDE-Based Methods, Enhancing PDEs, Neighborhood filters, Non-local Means algorithm.

Unit 3 : The Segmentation Problem (10 Periods)
The Mumford and Shah functional, Geodesic Active Contour and the Level set Method

Unit 4 : Image Classification (10 Periods)
Level-Set Approach for image classification, A variation model for image classification and restoration.

Unit 5 : Vector-Valued Images (10 Periods)
An extending notion of gradient, The Energy Method, PDE-Based Methods.

Total 42 Periods

Key Text: Gilles Aubert, Pierre Kornprobst, "Mathematical Problems in Image Processing" 2nd Ed, Springer Chapters 2, 3, 4, 5.4 and 5.5 from Key Text.

HCI 4(P) Practicals: ADVANCED TOPICS IN IMAGE PROCESSING

(1 Credit)

Course Objectives:

The course introduces the students to practical aspects of advanced treatment of the various tools of the area of Image Processing such as image restoration, segmentation and classification.

Course Outcomes: Upon completion of the lab course the student will

- Know how to make a basic implementation for solving the PDEs that emerge from the variational formulations

Syllabus:

Algorithms/Exercises from different units in the syllabus pertaining to image restoration, segmentation and classification will be implemented in Lab.

Key Text: Gilles Aubert, Pierre Kornprobst, "Mathematical Problems in Image Processing", 2nd Ed, Springer.

* * *

HCI 5 VIDEO PROCESSING

(3 Credits) (42 Periods)

Course Objectives:

In this course students will gain a broad overview of video technology, an understanding video analysis in frequency domain, get the basics of human visual systems, video as a sampled data set from space-time continuum in the real-world, video sampling rate conversion, video modeling, 2D and 3D motion estimation algorithms.

Course Outcomes:

- Student will have the understanding of lattice theory and sampling over lattice as applied to video signals.
- 2D motion models, Optical flow, pixel-based, mesh-based, motion estimation, global and region-based motion estimation.
- 3D motion estimation algorithms based on feature-based approach, direct and iterative type estimation.

Course Syllabus:

Unit 1: Introduction (5 Periods)

Video Formation and Representation, Analog and Digital Video

Unit 2: Video Sampling (8 Periods)

Basics of Lattice theory and Sampling over Lattices, Sampling video signals, Rate Conversion

Unit 3: Video Modeling (9 Periods)

Camera Model, Illumination Model, Object Model, Scene Model, 2-D Motion Models

Unit 4: 2D-Motion Estimation (10 Periods)

Optical Flow, Pixel Based Motion Estimation, Mesh Based Motion Estimation, Global Motion Estimation, Region-based Motion Estimation

Unit 5: 3D-Motion Estimation (10 Periods)

Feature Based Motion Estimation, Direct Motion Estimation, Iterative Motion Estimation

Total 42 Periods

Reference Text: Yao Wang, Jorn Ostermann and Ya-Qin Zhang, "Video Processing and Communications", 2002, Prentice Hall, Chapters 1- 7 from Key Text

Suggested Readings:

1. Yu-Jin Zhang, Advances in Image And Video Segmentation, IRM Press (May 2, 2006)
2. John W. Woods, Multidimensional Signal, Image, and Video Processing and Coding, Academic Press (March 13, 2006)

* * *

HCI 5(P) Practicals: VIDEO PROCESSING

(1 Credit)

Course Objective: Course introduces hands on experience through implementation of 2D and 3D motion estimation algorithms.

Course Outcome: At the completion of the course a student will have the knowledge of applying 2D and 3D motion estimation algorithms for video processing.

Syllabus:

- Pixel-based Motion Estimation
- Block-Matching Motion Estimation using Exhaustive Search, Fast Algorithms and Binary feature Matching.
- 3D motion estimation for known and unknown shapes using feature-based, direct and iterative approaches

Reference Text: Yao Wang, Jorn Ostermann and Ya-Qin Zhang, **"**Video Processing and Communications", 2002, Prentice Hall.

* * *

STREAM IV: THEORETICAL COMPUTER SCIENCE

TCS 1 ADVANCED ALGORITHMS

(3 Credits) (42 Periods)

Course Objectives:
The course handles problems that are NP-Complete and NP-Hard through approximation and randomization. Broadly speaking, the approximate algorithms only provide a suboptimal solution and take polynomial time in the input size. The course also provides tools for performing a probabilistic analysis of both approximate and randomized algorithms.

Course Outcomes: At the completion of the course the student will be able to

- Know how to design an approximate algorithm by inventing the necessary heuristic.
- Know how to convert a LP problem solved using simplex method into an approximate version.
- Understand Matroid and design greed algorithm.
- Design randomized algorithms.
- Design and analyze randomized data structures.

Unit 1: Preliminaries 7 (Periods)
Complexity classes-Lower bounding OPT – Well Characterized problems – MinMax relations - Chernoff bounds – The Minmax principle – Randomness and Non-Uniformity – Occupancy problems – Two point sampling – Stable marriage problem – Coupon Collector's problem

Unit 2: Approximate Algorithms 12 (Periods)
Matroid and greedy methods - Min cut algorithm – Las Vegas – Monte Carlo – Set Cover – Greedy algorithm – LP duality – Dual fitting – Rounding – Primal Dual Schema, Knapsack – pseudo-polynomial time algorithm – FPTAS

Unit 3: Randomized Algorithms 15 (Periods)
Probabilistic Recurrence - Randomized selection - Delaunay Triangulation- Minimum Spanning Trees - Counting Problems

Unit 4: Advanced data structures 8 (Periods)
Fundamental data structuring problem – Random Treaps - Skip lists – Hash tables with O(1) search time

Total 42 Periods

Reference Books:

1. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press
   [Chapters : 1, 2, 3, 4 (only 4.1), 8, 9(only 9.6), 10(only 10.3)]

2. Vijay V. Vazirani, Springer, Approximate Algorithm;
   [Chapters: 1, 2, 8, 12, 13, 14, 15, 28]

3. Thomas H. Cormen, Ronald L. Rivest and Clifford, Introduction to Algorithms, 2nd edition, MIT Press and McGraw-Hill, 2001.
   [Chapter: 16]

TCS 1(P) Practicals: ADVANCED ALGORITHMS

(1 Credit)

Course Objectives:
Algorithms/Exercises from different units in the syllabus will be implemented in Lab. The language of choice is Python.

Course Outcomes: At the completion of the lab course the student will be able to

- Know how to design an approximate algorithm by inventing the necessary heuristic.
- Know how to convert a LP problem solved using simplex method into an approximate version.
- Understand Matroid and design greed algorithm.
- Design randomized algorithms.
- Design and analyze randomized data structures.

Syllabus:

- write an approximate algorithm for a given NP-Complete problem.
- write an approximate version of the LPP.
- Code and test the performance of randomized algorithms
- Code and test the performance of randomized data structures.

Reference Books:

- Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms; Cambridge University Press
- Vijay V. Vazirani, Approximate Algorithm; Springer
- Thomas H. Cormen, Ronald L.Rivest and Clifford, Introduction to Algorithms, 2nd edition, MIT Press and McGraw-Hill, 2001.

* * *

TCS 2 CRYPTOGRAPHY (3 Credits) (42 Periods)

Course Objectives:
To provide foundations in cryptography for those students interested in pursuing cryptography, data privacy and network security.

Course Outcome: The students will be able to
- Understand the basics of cryptography
- Appreciate the philosophy behind symmetric and asymmetric key cryptography.
- Know when to and how to use public key primitives.

Course Syllabus:

Unit 1 : Introduction (2 periods)
OSI Security Architecture – Security Attacks - Security services – Security Mechanisms – A Model for Network Security

Unit 2 : Classical Cryptography Techniques (3 periods)
Symmetric Cipher Model-Substitution Techniques – Transposition Techniques – Rotor Machines – Steganography

Unit 3 : Block Cipher and the Data Encryption Standards (4 periods)
Block Cipher Principles – The Data Encryption Standards – The Strength of DES – Differential and Linear Cryptanalysis – Block Cipher Design Principles

Unit 4 : Finite Fields (3 periods)
Groups, Rings and Fields - Modular Arithmetic – The Euclidean Algorithm – Finite Fields of the Form GF(p) – Polynomial Arithmetic – Finite Fields of the Form GF(2^n)

Unit 5: Advanced Encryption Standard (4 periods)
Evaluation Criteria for AES – The AES Cipher

Unit 6 : More on Symmetric Ciphers (3 periods)
Multiple Encryption and Triple DES – Block Cipher Modes of Operation – Stream Ciphers and RC4

Unit 7 : Confidentiality Using Symmetric Encryption (4 periods)
Placement of encryption functions – Traffic Confidentiality – Key Distribution – random number Generation

Unit 8 : Introduction to Number Theory (3 periods)
Prime Number – Fermat's and Euler's Theorems – Testing for Primality – The Chinese Reminder Theorem – Discrete Logarithm Problem

Unit 9 : Public-Key Cryptographic and RSA (4 periods)

Principles of Public-Key Cryptosystems – The RSA algorithms - Key Management-Diffie-Hellman key Exchange - Elliptic Curve Architecture and Cryptography

Unit 10: Message Authentication and Hash Functions (4 periods)

Authentication requirements – Authentication functions – Message Authentication Codes – Hash functions – Security of Hash functions and MACs

Unit 11: Hash and MAC Algorithms (4 periods)

Secure Hash algorithm – Whirlpool – HMAC – CMAC

Unit 12: Digital Signatures and Authentication Protocols (4 periods)

Digital Signature – Authentication Protocols – Digital Signature Standards

Total 42 periods

Reference Text:

1. William Stallings, Cryptography and Network Security - Principles and Practices, Prentice Hall of India, 4th Edn, 2003.

[Chapters : 1 to 13 (except Recommended readings and Appendices from all chapters)]

Suggested Readings:

1. Atul Kahate, Cryptography and Network Security, Tata McGraw -Hill, 2003.
2. Bruce Schneier, Applied Cryptography, John Wiley & Sons Inc, 2001.

TCS 2(P) Practicals: CRYPTOGRAPHY (1 Credit)

Course Objective: To provide hands-on sessions for some cryptography primitives and attacks.

Course Outcome: The students will be able to implement some cryptography primitives and attacks.

Suggested Assignments:

- Substitution and transposition ciphers.
- Block ciphers, modes of operation and stream ciphers.
- Hash functions and Message authentication codes.
- Public cryptographic primitives like RSA.
- Digital signatures.

Reference Text:

1. William Stallings, Cryptography and Network Security - Principles and Practices, Prentice Hall of India, 4th Edn, 2003.

* * *

STREAM V: COMPUTER SYSTEMS

CS 1 COMPILER DESIGN (3 Credits) (42 Periods)

Course Objectives:

- Elicit the theory behind how a high level Programme gets translated to a Programme that machine can understand
- Clearly distinguish between the parts of the compiler that depend on the target machine and that do not depend on the target machine
- Introduce simple optimizations that compiler can perform to make the translated code more efficient in terms of space and time
- Supplement theory with existing tools like LEX and YACC to automate the lexical and syntax analysis phases of the compiler

Course Outcomes: At the end of this course, students should be:

- able to depict the lexical analysis phase with finite automata and convert the same to a practical program
- able to understand syntax analysis phase with parse trees and pushdown automata and convert the context free grammar to a practical Programme using top down parsing
- able to appreciate the sound theory behind parsing through bottom up parsing
- able to write attribute grammars for semantic analysis including type checking, type equivalence etc., and convert them to practical programs
- able to generate 3 address code or P-code as intermediate code
- able to appreciate the applications of compilation in other domains like natural language processing

Unit 1: Introduction - Why compilers? - Programs related to compiler - The Translation Process - Major Data structures in a compiler - Boot strapping and porting 2 periods

Unit 2: Scanning - The scanning process - Regular expressions - Finite automata - Regular expressions to DFA 5 periods

Unit 3: Context free grammars and Parsing - The Parsing process - CFG - Parse trees and Abstract Syntax Trees - Ambiguity 2 periods

Unit 4: Top-Down Parsing - Recursive descent parsing - LL(1) Parsing - First and Follow sets- Error recovery 6 periods

Unit 5: Bottom-Up Parsing - Overview - LR(0) parsing - SLR(1) parsing - LR(1) and LALR(1) parsing - Error recovery 6 periods

Unit 6: Semantic analysis - Attribute Grammar - Algorithms for attribute computation - Symbol table - Data types and type checking 10 periods

Unit 7: Runtime environments - Fully static environment- stack-based environment - Fully dynamic environment - Parameter passing mechanisms 5 periods

Unit 8: Code Generation - Intermediate code and data structures - Basic techniques - Code generation for data structure references - Code generation for control statements and logical expressions - Code generation for functions and procedure calls 6 periods

Total 42 periods

Reference Text:

Kenneth C. Louden, Compiler Construction: Principles and Practice, Cengage Learning Publishers, Indian Edition, 1997

[Chapters: 1.1-1.6, 2.1-2.4, 3.1 – 3.4, 4.1 – 4.3, 4.5, 5.1, 5.1 – 5.4, 5.7, 6.1-6.4, 7.1 – 7.5, 8.1 – 8.5]

Suggested Reading:

1. V. Aho, Ravi Sethi And J.D. Ullman, Compilers: Principles, Techniques And Tools, Addison Wesley Publishing Company, 2nd edition, 1986.
2. Charles N. Fischer, Ronald K. Cytron, Richard J. LeBlanc, Jr., Crafting A Compiler, Addison-Wesley, 2010.
3. Allen. I. Holub, Compiler Design in C, Prentice Hall of India, Eastern Economy Edition, Second Indian Reprint, 1993.

CS 1 (P) Practicals: COMPILER DESIGN (1 Credit)

Course Objectives:

- Make students implement a compiler for a simple language
- Make students implement auto tools for scanning and parsing based on the theory studied from the associated theory course

Course Outcomes:

At the end of this course, students will have the practical knowledge of implementing and testing some important concepts learned in the theory course.

Suggested exercises:

- design a simple language as a context free grammar
- implement analysis phase of the compiler for the simple language
- build auto tools for lexical and syntax analysis
- Implement intermediate code generation for the target machine, assuming availability of a simulator for a hypothetical target machine like SIC or SIC/XE.

Reference Text:

Kenneth C. Louden, Compiler Construction: principles and Practice, Cengage Learning Publishers, Indian Edition, 1997

* * *

CS 2 Embedded Computing (3 Credits) (42 Periods)

Course Objectives:

- To introduce students to the modern embedded systems and to show how to understand and program such systems using a concrete platform built around a modern embedded processor like the Intel ATOM or ARM processor based development boards & WinCE.

Course Outcome: After completing the course, one should be able to

- Describe the differences between the general computing system and the embedded system, also recognize the classification of embedded systems
- Understand and explain the architecture of a typical embedded processor and its programming aspects (assembly Level)
- Understand and explain interrupts, hyper threading and software optimization.

Course Syllabus:

Unit 1: An Overview of Embedded Computing (2 periods)

Introduction to embedded systems; characteristics, challenges; design process.

Unit 2: Embedded Hardware Fundamentals (10 periods)

Introduction to Microcontrollers, Microprocessors, DSPs; ARM & SHARC processors; I/O programming; execution modes; ARM & SHARC buses; CPU performance & power consumption; data compression example; I/O devices, component interfacing, design, development debugging and testing. Types of memory.

Unit 3: Embedded Software and Platforms (14 periods)

Interrupts & ISRs, Embedded OSs, Development tools and programming languages. Real time systems: Memory management, Scheduling, Utilization. Real time OS's: Desktop OS Vs RTOS, BSPs, Task management, Race conditions, Priority inversion, ISRs & Scheduling, Inter-Task Communication, Timers and other RTOS services.

Unit 4: Programming Exercises (16 periods)

Overview of: ARM processor based development boards, WinCE, RT-Linux. Simple programming exercises using C and assembly programming. Hint: use the examples discussed in Book1 & Reference 1.

Total 42 periods

Text Books:

1. Wayne Wolf, Computer as Components, Morgan Kaufmann Pub. Indian Edition 2008. [Chap. 1.1-1.4, 2.3,2.4,3.2, 3.6-3.8, 4.1-4.8]

2. Sriram V. Iyer, Pankaj Gupta, Embedded Real Time Systems Programming, Tata McGraw-Hill Pub. Co. Ltd, 2004. [Chaps 2, 3.2, Ch 4 to Ch 7]

Reference Books:

1. Raj Kamal, Embedded Systems, Hill Pub. Co. Ltd, 11th print 2007. [Chaps 1-5, Appendix G]

2. David E. Simon, An Embedded Software Primer, Pearson Education, 2007. [Chps 5-10]

3. Programming for Embedded Systems, Dream Software Team, WILEY dreamtech India Ltd. 2005. [has lots of excellent case studies].

CS 2(P) Practicals: Embedded Computing (1 Credit)

Objectives: To provide practical experience through lab exercises from some units in the syllabus.

Outcomes: At the end of the lab course , the student will be able to understand the characteristics and functioning principles of Microcontrollers, Microprocessors, DSPs such as ARM & SHARC processors. He/she will be able use some development boards for programming.

Syllabus:

ARM processor based development boards&WinCE, RT-Linux will be used for programming exercises using C and assembly programming. Examples discussed in Book1 & Reference 1 of theory course may be used for this lab.

* * *

CS 3 Advanced Programming in the UNIX Environment

(2 Credits) (28 Periods)

Course Objectives:

This course will introduce students to the basic and advanced features of the Unix Operating System. Features like, OS File system, Events generation and handling, Processes and their environment, Inter-process communication mechanisms, etc. Students with C background will code/Programme and familiarize with programmatic perspective of Unix Operating system.

They will perform some basic to advanced coding to customize and interact with underlying OS. Students are introduced to Unix OS library and utilize them to accomplish certain tasks with system level calls and features.

Course Outcomes: At the completion of the course the student will be able to

- Utilize Unix OS system Library and the C Standard Library to access/manipulate the file system, internal data structures, etc.
- Understand and apply Multithreading concepts, Inter-process communication mechanisms like, Pipe, FIFO, Shared Memory, Semaphores.
- Understand and implement concepts to create processes, child processes, data structures used and invoke system libraries to manipulate process.
- Code to generate the events, and handle different types of events generated by Kernel etc.

Course Syllabus:

Unit 1 : Introduction (4 Periods)

UNIX System Overview, UNIX Architecture Files and Directories, Input and Output, Error Handling, Signals, Time Values, System Calls and Library Functions

Unit 2 : File I/O (6 Periods)

File Descriptors, open and open at Functions, create Function, close Function, lseek Function, read Function, write Function, I/O Efficiency, File Sharing, dup and dup2 Functions sync, fsync, and fdatasync Functions, fcntl Function

Unit 3: Files and Directories (6 Periods)

stat, fstat, fstatat, and lstat Functions, File Types, File Access Permissions, Ownership of New Files and Directories, chmod, fchmod, and fchmodat Functions, chown, fchown, fchownat, and lchown Functions, link, linkat, unlink, unlinkat, and remove Functions, Creating and Reading Symbolic Links, Reading Directories, chdir, fchdir, and getcwd Functions, Device Special Files

Unit 4: Process Control (6 Periods)

fork Function, vfork Function, exit Functions, wait and waitpid Functions, waitid Function, wait3 and wait4 Functions, Race Conditions, exec Functions, Interpreter Files, system Function

Unit 5: Signals (6 Periods)
signal Function, Unreliable Signals, Interrupted System Calls, Reentrant Functions, SIGCLD Semantics, kill and raise Functions, alarm and pause Functions, sigprocmask Function, sigpending Function, sigaction Function, sigsetjmp and siglongjmp Functions, sigsuspend Function, sleep, nanosleep, and clock_nanosleep, sigqueue Function

Total (28 Periods)

Reference Text: Advanced Programming in the UNIX Environment Third Edition by W. Richard Stevens Stephen A. Rago

CS 3(P) Practicals: Advanced Programming in the UNIX Environment

(2 Credits) (28 periods)

Course Objectives:
This course will introduce students to the hands-on training for the advanced features of the Unix Operating System.

Course Outcomes: At the completion of the course the student will have the practical knowledge of

- How to access/manipulate the file system, internal data structures.
- Implementing Multithreading, Inter-process communication mechanisms like Pipe, FIFO, Shared Memory, Semaphores.
- Implementing concepts to create processes, child processes, data structures used and invoke system libraries to manipulate process.
- Hands on exercises on Files and File system like creating a hole in a file, retrieving and changing File meta data and modify access permissions, accessing data structures of internal File system.
- Coding on creating processes and child processes and assigning tasks and communication between parent and children.

Reference Text:
W. Richard Stevens Stephen A. Rago, Advanced Programming in the UNIX Environment Third Edition.

* * *

CS 4 Programming for performance

(2 Credits) (28 Periods)

Course Objective: To introduce the mind-set of performance driven programming which relies on the Computer Architecture knowledge.

Course Outcome: at the end of the course, the student will be able to analyze the performance aspects of a program. He will be able to think in a way which will improve the performance of a program, not merely correctness.

Course Syllabus:

Unit 1: Introduction (4 Periods)
basic concepts, cost/performance analysis, Valgrind and gdb, Parallel debuggers.

Unit 2: Architecture/Microarchitecture (6 Periods)
Operational intensity, Core 2/Core i7, Compute Core Optimizations
In-core optimizations (ILP - pipelining, superscalar etc. branch predictions etc.)
Assembly level optimizations, Profiling tools: perfexpert, perf, How to write branchless code,

Unit 3: Benchmarking (6 Periods)
Benchmarking (Issues with accurately timing a code)
Nano level benchmarking: Discuss X-ray paper
Timer granularity. Compare c-time, get time of day and rdtsc etc.

Unit 4: Vectorization (6 Periods)
SIMD vectorization, SIMD programming (scalar dot product),
How to write code so as to make compiler generate effective SIMD code

Unit 5: Memory Locality Optimizations (6 Periods)
Locality (Memory specific optimizations)
Prefetching, caching, cache blocking, register blocking etc.
Loop specific optimizations., membench (and memory mountain),
versions of MMM using perfexpert, perf and papi, Optimize Matrix Transpose

Total (28 Periods)

Reference Material:
Much of the material used in this course is taken from the courses taught by distinguished professors: Prof. Markus Puschel (ETH), Prof. Maria Garzaran (UIUC), Prof. Keshav Pingali (UT), Prof. Charles E. Leiserson (MIT), Prof. Richard Vudua (GaTech), Prof. James Demmel and Prof. Dan Garcia (UCB), Prof. Saday (Ohio).

CS 4(P) Practicals: Programming for performance

(2 Credits)

Course Objective: To introduce the mind-set of performance driven programming which relies on the Computer Architecture knowledge through hands-on training.

Course Outcome: At the end of the course, the student will be able to analyze the performance aspects of a program. He will be able to think in a way which will improve the performance of a program, not merely correctness.

Syllabus:
Relevant exercises from different units in the syllabus will be implemented in Lab.

***

CS 5 Operating Systems (3 Credits) (42 periods)

Course objectives: The objective of the course is to provide basic knowledge of computer operating system structures and functioning.

Course outcomes: Upon completing the course, the student is expected to

- Explain the basic structure and functioning of operating system.
- Identify the problems related to process management and synchronization as well as apply learned methods to solve basic problems.
- Understand the cause and effect related to deadlocks and is able to analyse them related to common circumstances in operating systems.
- Explain basics of memory management, the use of virtual memory in modern operating systems as well as the structure of the most common file-systems.

Course Syllabus:

Unit 0: Introduction: Operating system (OS) concepts – System Calls – OS structure
6 Periods

Unit 1: Process Management: Processes – Threads – Inter process communication – Scheduling – Processes in MINIX 10 Periods

Unit 2: I/O: I/O hardware – I/O software – Deadlocks – I/O in MINIX 10 Periods

Unit 3: Memory Management: Swapping – Virtual Memory – Paging – Segmentation – Process Manager in MINIX 10 Periods

Unit 4: File System: Files – Directories – File System Implementation - MINIX File System
6 Periods

Total 42 Periods

Reference text book

1. The MINIX book - Operating Systems – Design and Implementation, Andrew S. Tanenbaum, Third Edition, Pearson Education, 2006.
[Chapters: 1, 2(2.1 – 2.5), 3(3.1-3.4), 4(4.1-4.7), 5(5.1 – 5.3, 5.6)]

Suggested Readings:

1. Andrew S. Tanenbaum, Modern Operating Systems, III Edn, Pearson Education, 2001.

2. Silberchatz A & Gallvin, Operating System Concepts, VII Edn, Addison Wesley, 1997.

3. William Stallings, Operating Systems, III Edn, Pearson Education, 2001.

* * *

CS 5(P) Practicals: Operating Systems (1 Credit)

Course objectives: The objective of this lab course is to provide hands on experience in modifying the MINIX source code and rebuilding the same to see the effect.

Course outcomes:

Student will have the experience of going through large code base. He/She will develop debugging, building and navigation skills. Student will also gain experience in using certain tools for handling system level coding.

Syllabus:

Exercises from key text book of CS5 can be experimented with in Lab. For example, changing the existing scheduling algorithm or memory management scheme.

* * *

STREAM VI: MULTI-CORE AND PARALLEL COMPUTING

MPC 1 PARALLEL NUMERICAL LINEAR ALGEBRA

(3 Credits) (42 Periods)

Course Objective: To expose and develop understanding on the following:

- Performance Analysis
- Direct Solutions of sparse linear systems
- Projections method, Iterative methods for linear systems of equations
- Eigenvalue problems

Course Outcome: To develop proficiency in designing parallel algorithms for the following:

- Direct and Iterative methods for Linear systems of equations
- Krylov subspace methods
- Performance Measurement of such systems

Course Syllabus:

Unit 1: Overview - Implementation–Performance analysis- Modeling – Measurements. (4 periods)

Unit 2: Building blocks in Linear Algebra–Direct solution of sparse linear system (6 periods)

Unit 3: Krylov subspaces – projection (8 periods)

Unit 4: Iterative methods for linear systems (8 periods)

Unit 5: Preconditioning and parallel preconditioning –Linear eigen value problem – Generalized eigenvalue problem. (8 periods)

Unit 6: Implementation of selected methods in a specific parallel programming platform. (2 periods)

Total 42 periods

TEXTBOOK

1. Jack J Dongarra, Lain S Duff, Danny C Sorrenson, H. A .Vander Verst, Numerical Linear Algebra for High Performance Computers, ACM Portal, 1998 (ISBN: 0898714281)

REFERENCE BOOKS

1. Martin Gloubitsky, Michael Dellsitz, Linear Algebra and Differential Equations using MATLAB, Brooks / Cole Publishing Co, ACM Portal, 1999.

MPC 1(P) Practicals: PARALLEL NUMERICAL LINEAR ALGEBRA

(1 Credit)

Relevant exercises from different units in the syllabus will be implemented in Lab to strengthen the theoretical concepts learned in the theory course.

* * *

MPC 2 MULTI CORE COMPUTING

(3 Credits) (42 Periods)

Course Objective: To expose and develop understanding on the following:

- Parallel computing, Multi Core Architecture, Multi-Threading
- Communication operations
- Analysis of Parallel Programs
- Shared Memory and Multi thread programming

Course Outcome: To develop basic skill in the following:

- Multi Core Architecture, Threading and performance
- Parallel Algorithms Models
- Load Balancing
- Communication calls like Broadcast, Scatter, Gather, Reduction, Shift
- Open MP and p-Thread

Course Syllabus:

Introduction to Multi-Core Architecture: (6 periods)

Motivation for Concurrency in Software - Parallel Computing Platforms - Parallel Computing in Microprocessors - Differentiating Multi - Core Architectures from Hyper-Threading Technology - Multi-threading on Single-Core versus Multi -Core Platforms - Understanding Performance

Principles of Parallel Algorithm Design: (5 periods)

Preliminaries - Decomposition Techniques – Characteristics of Tasks and Interactions – Mapping Techniques for Load Balancing-Methods for containing Interaction overheads – Parallel Algorithm Models

Basic Communication Operations: (5 periods)

One – to - All Broadcast and All–to-All Reduction – All–to-All Broadcast and Reduction – All Reduce and Prefix Operations - Scatter and Gather – Circular Shift

Analytic modeling of Parallel Programs: (5 periods)

Source of Overhead in Parallel Programme – Performance Metrics –for Parallel System – Scalability of Parallel System – Asymptatic Analysis of Parallel Program.

System Overview of threading: (6 periods)

Defining Threads – System view of Threads – Threading above OS – Threading inside OS – Threading inside Hardware – Application of Programming Models and Threading – Virtual Environment – Runtime Virtualization – System Virtualization.

Programming using the Message – Passing Paradigm: (6 periods)
Principles of Message Passing Programme – Basic Building Blocks – Topologies and Embedding – Collectiv Communications and Computation Operations

Programming shared address space platforms: (5 periods)
Thread basics – Synchronization Primitives in PThreads-Controlling Thread and Synchronization Attributes – Thread Cancellation – OpenMP: Standard for Directive Based Parallel Programming

Algorithms: (4 periods)
Dense Matrix - Algorithms - Sorting Algorithms - Graph Algorithms - Search Algorithms for Discrete Optimization Problems

Total 42 periods

TEXT BOOKS

1. Shameem Akhter and Jason Roberts, Multi-core Programming (increasing performance through software multi-threading), Intel Press(2006)
2. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Second Edition, Addison-Wesley, 2003

MPC 2(P) Practicals: MULTI CORE COMPUTING (1 Credit)

Relevant exercises from different units in the theory course will be implemented in Lab.

* * *

MPC 3 HIGH PERFORMANCE EMBEDDED COMPUTING

(3 Credits) (42 Periods)

Course Objectives:

To develop a student's ability in embedded computing to the level of designing high performance embedded systems using Multiprocessor Design Techniques and Embedded Multiprocessor Software.

Course Outcomes:

By the end of the module the student should be able to:

- Apply the more advanced features of architectures in high performance embedded systems design.
- Explain Multiprocessor Design and Interconnection Network for a High Performance Embedded systems
- Integrate a hardware accelerator with a processor and design the necessary software and hardware communication infrastructure

Course Syllabus:

Embedded Computing: The Landscape of High - Performance Embedded Computing Design Methodologies - Models of Computation, Reliability, Safety, and Security

CPUs: Comparing Processors - RISC Processors and Digital Signal Processors Parallel Execution Mechanisms - Variable-Performance CPU Architectures

Programs: Code Generation and Back - End Compilation - Memory -Oriented Optimizations - Programme Performance Analysis - Models of Computation and Programming

Processes and Operating Systems: Real -Time Process Scheduling-Languages - and Scheduling Operating System Design and Verification

Multiprocessor Architectures: Embedded Multiprocessors - Multiprocessor Design Techniques - Interconnection Networks - Physically Distributed Systems and Networks - Multiprocessor Design Methodologies and Algorithms

Multiprocessor Software: Embedded Multiprocessor Software - Real -Time Multiprocessor Operating Systems - Services and Middleware for Embedded Multiprocessors - Design Verification

Hardware / Software Co-Design: Design Platforms– Performance Analysis Hardware. Software Co-Synthesis Algorithms -Hardware/Software Co-Simulation

Reference Book

1. Wayne Wolf, HIGH - PERFORMANCE EMBEDDED COMPUTING - Architectures, Applications, and Methodologies, 1st Ed, Elsevier Pub 2006.

MPC 3(P): Practicals: HIGH PERFORMANCE EMBEDDED COMPUTING

(1 Credit)

Course Objectives:

To develop a student's ability in practical aspects of embedded computing through simulation experiments

Course Outcomes:

By the end of the module the student should be able to demonstrate through simulations, the design principles of architectures in high performance embedded systems design.

Syllabus:

Relevant exercises from different units in the syllabus will be implemented in Lab.

Reference Book

1. Wayne Wolf, HIGH - PERFORMANCE EMBEDDED COMPUTING - Architectures, Applications, and Methodologies, 1st Ed, Elsevier Pub 2006.

* * *

MPC 4 HIGH PERFORMANCE COMPUTING WITH ACCELERATORS

(2 Credits) (28 Periods)

Unit 1: Data Parallel Computing (6 periods)
Data parallelism, CUDA Kernel, Device Global memory, Function declaration, Kernel launch.

Unit 2: Scalabale Parallel execution (8 periods)
CUDA Thread organization, Synchronization, Scheduling and Latency Tolerance. Memory and data Locality, Tiling for reduced memory Traffic, Boundary checks, Global Memory bandwidth, Warp, Dynamic Partitioning of Resources, Thread Granularity.

Unit 3: Parallel Algorithm Patterns (8 periods)
Convolution, Prefix Sum, Parallel Histogram, Sparse matrix Computation, Merge Sort, Graph Search.

Unit 4: CUDA Dynamic Parallelism and **Open ACC** (6 periods)

Case Study: Machine Learning.

Total: 28 Periods

Note: Depending on the circumstances, OpenCL could be used in place of CUDA (OpenCL is another language similar to CUDA which is gaining importance recently).

TEXT BOOKS

1. Programming Massively Parallel processors, A hands on Approach, Third Edition, by David B Kirk and Wen-mei W. Hwu, Elsevier Morgan Kauffman, 2017.

REFERENCES:

1. CUDA by Example: An Introduction to General-Purpose GPU Programming, by Jason Sanders and Edward Kandrot, Addison Wesley Professional, 1st edition, 2010.
2. GPU Computing Gems by Wen-mei W. Hwu, Emerald Edition, Morgan Kauffman Publishers, 2011.
3. Heterogeneous Computing with OpenCL by Benedict R. Gaster, et al., Morgan Kaufmann, 2012.

MPC 4(P)
Practical : HIGH PERFORMANCE COMPUTING WITH ACCELERATORS

(2 Credits)

Unit 1:
GPU programming Model: introduction to CUDA, CUDA Execution Model.
Lab 0: Work through simple CUDA example, Synchronization, CUDA debugging and profiling tools.

Lab 1: Debug and profile a simple kernel before optimizing it.

Unit 2:
Lab 2: programming assignment of simple and tiled matrix multiplication in.
Test 1: Quiz based on descriptive questions.

Unit 3:
Lab 3: programming assignment of optimizing reduction tree. (memory specific optimizations)
Lab 4 (optional): Optimal binding of CPU cores and GPUs to a process.
Lab 5 (optional): Optimal mapping of CPU cores to GPUs in MPI-CUDA based applications.

Unit 4:
Course Project: This is a project intensive course, where in the student groups will apply the performance optimization techniques learned here to a chosen application.

Note: Depending on the circumstances, OpenCL could be used in place of CUDA. (OpenCL is another language similar to CUDA which is gaining importance recently).

TEXT BOOK

1. Programming Massively Parallel processors, A hands on Approach, Third Edition, by David B Kirk and Wen-mei W.Hwu, Elsevier Morgan Kauffman, 2017.
2. Programming Massively Parallel processors, A hands on Approach, Second Edition, by David B Kirk and Wen-mei W.Hwu, Elsevier Morgan Kauffman, 2012.
3. Annual nVIDIA GTC Conference Presentations on Performance Optimizations and toolset.

REFERENCES:

1. CUDA by Example: An Introduction to General-Purpose GPU Programming, by Jason Sanders and Edward Kandrot, Addison Wesley Professional, 1st edition, 2010.
2. GPU Computing Gems by Wen-mei W. Hwu, Emerald Edition, Morgan Kauffman Publishers, 2011.
3. Heterogeneous Computing with OpenCL by Benedict R. Gaster, et al., Morgan Kaufmann, 2012

* * *

MPC 5 CLOUD COMPUTING (2 Credits) (28 Periods)

Course Objective: To expose and develop understanding on the following:

- HPC and Distributed Computing platforms
- Clusters, GRID
- Virtual Machines, Clusters and data Centers
- Cloud Architecture and Service Models
- Programming Models and Software environments
- IoT, online netwroking

Course Outcome: To develop basic skill in the following:

- Scalable computing over internet
- Design Principles of Clusters
- Performance, Security and energy efficiency of Distributed clusters and cloud
- Cloud service models
- Virtualization
- Cloud security and Trust management

Course Syllabus:

UNIT- 1 (8 Periods)

Distributed System Models and Enabling Technologies:: Scalable Computing over the internet - Technologies for Network-Based Systems -- System Models for Distributed and Cloud Computing -- Software Environments for Distributed Systems and Clouds -- Performance, Security, and Energy Efficiency.

Computer Clusters for Scalable Parallel Computing:: Clustering for Massive Parallelism -- Computer Clusters and MPP Architectures --Design Principles of Computer Clusters -- Cluster Job and Resource Management –

Case Studies of Top Supercomputer Systems --Tianhe-1A: The World Fastest Supercomputer in 2010 --XT5 Jaguar: The Top Supercomputer in 2009 -- IBM Roadrunner: The Top Supercomputer in 2008. (Cases may be changed with the availability of information from time to time regarding TOP supercomputers)

UNIT-2 (15 Periods)

Virtual Machines and Virtualization of Clusters and Data Centers -- Implementation Levels of Virtualization--Virtualization Structures/Tools and Mechanisms -- Virtualization of CPU, Memory, and I/O Devices -- Virtual Clusters and Resource Management -- Virtualization for Data-Center Automation.

Cloud Platform Architecture over Virtualized Data Centers -- Cloud Computing and Service Models -- Data-Center Design and Interconnection Networks -- Architectural Design of Compute and Storage Clouds -- Public Cloud Platforms: GAE, AWS, and Azure -- Inter-cloud Resource Management -- Cloud Security and Trust Management –

Service-Oriented Architectures for Distributed Computing -- Services and Service-Oriented Architecture -- Message-Oriented Middleware -- Portals and Science Gateways -- Discovery, Registries, Metadata, and Databases -- Workflow in Service-Oriented Architectures

Cloud Programming and Software Environments -- Features of Cloud and Grid Platforms -- Parallel and Distributed Programming Paradigms -- Programming Support of Google App Engine -- Programming on Amazon AWS and Microsoft Azure -- Emerging Cloud Software Environments

UNIT-3 (5 Periods)
Ubiquitous Clouds and the Internet of Things: Cloud Trends in Supporting Ubiquitous Computing -- Performance of Distributed Systems and the Cloud -- Enabling Technologies for the Internet of Things -- Innovative Applications of the Internet of Things --Online Social and Professional Networking

Total 28 periods

Text Book:
Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things.

Original English language edition copyright © 2012 by Elsevier Inc.

MPC 5(P) Practicals: CLOUD COMPUTING **(2 Credits)**

Course Objective: To expose and develop practical understanding on the topics studied in the theory course such as Virtual Machines, Clusters and data Centers, Cloud Architecture and Service Models.

Course Outcome: At the end of the course a student will have the basic skill in Scalable Computing over the internet - Technologies for Network-Based Systems, Design Principles of Clusters, Cloud service models, Virtualization, Cloud security and Trust management.

Virtual Machines and Virtualization of Clusters and Data Centers:
Cloud Platform Architecture over Virtualized Data Centers -- Public Cloud Platforms: GAE, AWS, and Azure -- Inter-cloud Resource Management -- Cloud Security and Trust Management.

Service-Oriented Architectures for Distributed Computing -- Services and Service-Oriented Architecture -- Message-Oriented Middleware -- Portals and Science Gateways -- Discovery, Registries, Metadata, and Databases -- Workflow in Service-Oriented Architectures
Cloud Programming and Software Environments -- Programming Support of Google App Engine -- Programming on Amazon AWS and Microsoft Azure

Reference Text:
Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Original English language edition copyright © 2012 by Elsevier Inc.

The text book will be supplemented with research papers and assignments & Projects designed for the course by the instructor.

* * *

MPC 6 Multi Processor Programming (2 Credits) (28 Periods)

Course Objectives: To expose and develop understanding on the following:

- Mutual exclusions, Concurrent objects, shared memory
- Consensus and Locks
- Synchronization
- Transactional Memory

Course Outcomes: To develop basic skill in the following:

- Threaded solution to exclusion
- Synchronization operators
- Various types of locking mechanisms
- Barriers and Transactional memory

Course Syllabus:

UNIT-I: Mutual Exclusion: Critical Section - 2Thread solution- Filter Lock - Lamport Bakery Algorithm - Bounded Time stamp - Concurrent objects – Shared Memory - Primitive Synchronization operators (4 Periods)

UNIT-II: Universality Consensus: Lock Free Universal Construction- Wait Free Universal Construction - Spin Locks and Contention: Queue Locks – Composite Locks- Hierarchical Locks - Monitor Locks and conditions -Readers-writers Locks (6 Periods)

UNIT-III: Linked Lists and role of Locking - Fine & Coarse grained synchronization, Optimistic Synchronization, Lazy Synchronization, Non-Blocking Synchronization. (6 Periods)

UNIT-IV: *Concurrent Queues – Concurrent stacks - Concurrent Hashing* - Skip Lists and Balanced Search (4 Periods)

UNIT-V: *Barriers* (4 Periods)

UNIT-VI: Transactional Memory (4 Periods)

Total (28 Periods)

Reference Text:

Maurice Herlihy, Nir Shavit, The Art of Multiprocessor Programming, Elsevier 2008. Morgan Kaufmann Publishers is an imprint of Elsevier. ISBN: 978-0-12-370591-4

MPC 6(P) Practicals: Multi Processor Programming

(2 Credits)

Course Objective: To expose and develop practical understanding on the following:

- Mutual exclusions, Concurrent objects, shared memory
- Consensus and Locks
- Synchronization
- Transactional Memory

Course Outcome: The student will have the practical skill in the following:

- Threaded solution to exclusion
- Synchronization operators
- Various types of locking mechanisms
- Barriers and Transactional memory

Syllabus:

Relevant exercises from different units in the syllabus will be implemented in Lab.

Reference Text:

Maurice Herlihy, Nir Shavit, The Art of Multiprocessor Programming, Elsevier 2008. Morgan Kaufmann Publishers is an imprint of Elsevier. ISBN: 978-0-12-370591-4

***

STREAM VII: SOFTWARE ENGINEERING

SE 1 OBJECT ORIENTED SYSTEM DESIGN

(3 Credits) (42 Periods)

Course Objectives:

Understanding the role played by the concept of "Objects" in computer science. Students are already aware of the ease in programming due to the concept of "object(s)". In this course a student also learns how even operating systems and software design can be understood better by this concept.

Course outcomes:

1) Learning object oriented design tools
2) Object oriented software design
3) System analysis based on object orientation

Course Syllabus:

INTRODUCTION

Overview of Object Oriented Systems Development - Object Basics: The object Model - Classes and Objects - Complexity - Notation - Process - Object types - Object state – Object - Oriented Systems Development Life Cycle.

OBJECT ORIENTED METHODOLOGIES

Rumbaugh methodology - Booch methodology – Jacobson methodology –Patterns – Frameworks – Unified approach – Unified Modeling Language - Use case – Class diagram – Interactive diagram – Package diagram – Colloboration diagram – State diagram– Activity Diagram.

OBJECT ORIENTED ANALYSIS:

Identifying use cases – Object analysis – Classification – Identifying object relationships – Attributes and methods.

OBJECT ORIENTED DESIGN

Design axioms – Designing classes – Access layer– Object storage – Object interoperability.

SOFTWARE QUALITY AND USABILITY

Designing interface objects–Software quality assurance– System usability–Metrics.

Reference Texts

1. Ali Bahrami, Object Oriented Systems Development, Irwin McGraw – Hill,1999.
2. Martin Fowler, UML Distilled, IInd Edn, PHI / Pearson Education, 2002.

Suggested Readings:

1. Grady Booch, Object Oriented Analysis and Design with Applications, IInd Edn, Benjamine Cummings, USA, 1994.
2. James R. Rumbaugh, Michael R. Blaha et al, Object Oriented Modeling and Design, Pearson Education Asia, 1991.
3. Betrand Meyor, Object Oriented Software Construction, IInd Edn, Prentice Hall PTR, New Jersey, 1997
4. Stephen R Schach, Introduction to Object Oriented Analysis and Design, Tata McGraw-Hill, 2003.
5. Tom Pender, UML 2 Bible, Wiley Publishing, Inc., 2005

SE 1(P) Practicals: OBJECT ORIENTED SYSTEM DESIGN

(1 Credit)

Course Objective: Use state diagrams, activity diagrams to model the functioning of a system. Use UML diagrams to depict the various divisions involved in making a function happen.

Course Outcomes:

1) Learn UML diagrams
2) Learn Activity diagrams
3) Learn Use case analysis
4) Learn collaboration diagrams.

Syllabus:

Relevant exercises from different units in the syllabus will be implemented in Lab.

Reference Texts

1. Ali Bahrami, Object Oriented Systems Development, Irwin McGraw – Hill,1999.
2. Martin Fowler, UML Distilled, IInd Edn, PHI / Pearson Education, 2002.

* * *

SE 2 WEB TECHNOLOGY **(3Credits)** **(45 Periods)**

Course Objective: This course will introduce the students to Web applications and their environment using Java technologies. HTTP protocol, HTML forms, web servers, containers, XML trees and parsing, JSPs, servlets, web services and other web jargon are familiarized.

Course Outcomes:

- Students would have good theoretical understanding about web app environment, Web services, and different technologies, frameworks pertaining to web applications.
- Good understanding about web services like SOAP, RESTful; XML parsing, HTML methods, application containers and their environment.

Course Syllabus:

Unit-1 Introduction (2 Periods)
The importance of web technology, Evolution of the Web, Basic concepts -TCP/IP, Ipv4 Vs Ipv6, Important Web components, App/Web servers, Internet who's who

Unit-2 Client Side components: (14 Periods)
Client Side Programming, Markup languages – HTML, Validators, HTML5 elements; Cascading Style Sheets (CSS), CSS Box Model, Manipulating behaviour – Java Script Language, Functions, Objects, Closures, Browsers and DOM, Event handling, Client side Ajax, Cookies (Client), Javascript security, Introduction to unobtrusive frameworks like jQuery. Lab Hands-on

Unit-3 Server Side Programming : (10 Periods)
Client side Vs Server side, Why Server? Java Servlet Architecture, Servlet Lifecycle, Web Application architecture, Session management, Server Cookies, Event handling on the server side, **JSP,** JSP Lifecycle, Expression language concepts, , Database connections / JDBC, Server side Ajax, Lab hands-on with Servlets / JSP

Unit-4 Server Side Business Layer (3 Periods)
Enterprise Java Beans (EJB-3), EJB versions, Session Beans, Entity Beans, Message driven beans, Java persistence API (JPA) concepts. Hands-on Lab

Unit-5 – Server Side Foundations of Interoperability & Standardization
(16 Periods)
Application to application messages, XML, Fundamental building blocks of XML, XML Parsing, DTD/Schema, XSL concepts, introduction to JSON
Web Services Introduction, Evolution of Web Services Technologies, Architecture – Basic Technology: SOAP (Simple Object Access Protocol) , WSDL, REST, Path to Service Oriented Architecture. Hands-on Lab

Total 45 Periods

Reference Text:

1. M Srinivasan, Web Technologies – Theory and Practice, Pearson, 2012. All Chapters

Suggested Readings:

1. Anders Miller, An Introduction to XML and Web Technologies, Michael Schwartzbach, Addison Wesley, 2006.
2. Michael P Papzoglou, Web Services: Principles and Technology, Pearson - Prentice hall, 2007.
3. Gustavo Alonso, Web Services: Concepts, Architecture and Applications, Fabio Casati , Harumi Kuno ,Vijay Machiraju , Springer – Verlag , 2004.

SE 2(P) Practicals: WEB TECHNOLOGY (1Credit)

Course Objective: This course will introduce the students to practical knowledge on developing simple Web applications and their environment using Java technologies.

Course Outcomes: At the end of the course, the student will have a basic knowledge of HTTP protocol, HTML forms, web servers, containers, XML trees and parsing, JSPs, servlets etc. Students would be able to code/develop and deploy a simple dynamic web application in a web application server using java, jsps, servlets, HTML, CSS, javascript, XML and connecting to backend database.

Course Syllabus:

Lab1: Create simple HTML page to submit a form using HTML, Cascading style sheets(CSS), and Javascript for form validation, handle few events using JQuery scripts, and use AJAX for dynamic content delivery. Understand HTTP protocol.

Lab2: Create JSPs for client side web requests and servlets to handle the requests and send JSP response back to web client. Use of cookies, understand the lifecycle of JSPs and Servlets; understand Request, Response and Session objects using a complete web transaction.

Lab3: Implement data layer using EJBs like Entitiy beans, Session Beans and Java Persistance API (JPA).

Lab4: XML parsing and JSON parsing using few available libraries.

Lab5: Implementing few Web services like SOAP (Simple Object Access Protocol) , WSDL, RESTful service using few existing frameworks.

Reference Text:

1. M Srinivasan, Web Technologies – Theory and Practice, Pearson, 2012. All Chapters

* * *

STREAM VIII: MATHEMATICAL METHODS IN COMPUTER SCIENCE

MMCS 1 Mathematical Methods in Image Processing

(3 Credits) (42 Periods)

Course Objectives:

The course familiarizes the students with the advanced mathematical tools necessary for the area called Image Processing. A special emphasis is given to the mathematical areas of functional analysis, partial differential equations and calculus of variations approach. Image Restoration is discussed as case study of the mathematical methods.

Course Outcomes: Upon completion of the course the student will

- Be given a deeper knowledge of Functional Analytic methods in Image Processing
- Be able to make a variational formulation (wherever possible) of a task in image processing
- Learn some of the modern algorithms for image restoration.
- Prepare for research skill associated with the domain of Image Processing
- Know how to make a basic implementation for solving the PDEs that emerge from the formulation

Course Syllabus:

Unit 1: Introduction: **(6 Periods)**

What is a Digital Image? Partial Differential Equations and Image Processing

Unit 2: Mathematical Preliminaries: **(18 Periods)**

Direct methods in the Calculus of Variations
Space of Bounded Variation functions
Viscosity solutions in PDEs
Curvature
Other classical results

Units 3: **Image restoration:** **(18 Periods)**

Image Degradation
The Energy Method
Regularization problem
PDE-Based methods: Nonlinear Diffusion,
Smoothing-Enhancing PDEs
Scale space theory

Total of Periods: **(42 Periods)**

Reference Text: Gilles Aubert, Pierre Kornprobst, Mathematical Problems in Image Processing, Springer; 1 edition (November 9, 2001)
[Chapters: 1, 2, 3.]

MMCS 1(P) Practicals: Mathematical Methods in Image Processing

(1 Credit)

Course Objectives: This lab based course is intended to train the students on mathematical tools necessary for Image Processing operations. A special emphasis is given to the mathematical areas of functional analysis, partial differential equations and calculus of variations approach.

Course Outcomes: Upon completion of the course the student will

- Know how to make basic implementation for solving the PDEs that emerge from the formulation for specific operation on image domain.

Relevant exercises from different units in the syllabus can be implemented in Lab.

Reference Text: Gilles Aubert, Pierre Kornprobst, Mathematical Problems in Image Processing, Springer; 1 edition (November 9, 2001)

* * *

MMCS 2 Numerical methods in Image Processing

(3 Credits) (42 Periods)

Course Objectives:

The course familiarizes the students with the advanced mathematical tools necessary for the area called Image Processing. A special emphasis is given to the mathematical areas of differential geometry and partial differential equations. Some of the recent advances in Image Processing such as Level Set Methods are discussed in detail.

Course Outcomes: Upon completion of the course the student will

- Be given a deeper knowledge of Geometric methods in Image Processing
- Be able to make a geometric formulation (wherever possible) of a task in image processing
- Learn some of the modern algorithms of image processing such as Level Set Methods.
- Prepare for research skill associated with the domain of Image Processing

Unit 1: Short introduction to calculus of variations, Short introduction to differential geometry	(8 Periods)
Unit 2: Curve evolution theory and invariant signatures	(8 Periods)
Unit 3: The Osher-Sethian level-set method	(7 Periods)
Unit 4: The level-set method: numerical considerations	(7 Periods)
Unit 5: Mathematical morphology, Distance maps and skeletons	(7 Periods)
Unit 6: Problem Solving	(5 Periods)
Total :	**(42 Periods)**

Reference Text: Ron Kimmel, M. Bronstein, A. Bronstein, Numerical Geometry of Images, Springer, 2003. [Chapters: 1 to 6]

MMCS 2(P) Practicals: Numerical methods in Image Processing

(1 Credit)

Course Objectives: To introduce hands-on training on some mathematical tools necessary for Image Processing.

Course Outcomes: Upon completion of the course the student will

- able to use the 3D Slicer software and be able to work with geometry of 3D images
- able to implement the Level Set Methods and apply the method for image segmentation

Syllabus:

Relevant exercises/concepts from different units in the syllabus can be implemented in Lab.

Reference Text: Ron Kimmel, M. Bronstein, A. Bronstein, Numerical Geometry of Images, Springer, 2003.

* * *

MMCS 3 Mathematical Methods for Data Mining

(3Credits) (42 Periods)

Course objectives: This course is concerned with data mining - that is, finding interesting and useful patterns in large data repositories. It aims to provide the student with conceptual and practical knowledge on important developments in data mining. Main objective of the course is to deliver the main concepts, principles and techniques of data mining so that the student will develop the confidence to analyse data of various forms, including transaction data, relational data and textual data.

Course outcomes:

- Demonstrate fundamental knowledge of data mining concepts and techniques.
- Apply the techniques of clustering, classification, association finding, feature selection and visualisation on real world data
- Apply data mining software and toolkits in a range of applications
- Set up a data mining process for an application, including data preparation, modelling and evaluation

Course Syllabus:

Unit 1: Introduction: Motivating Challenges, The Origins of Data Mining, Data Mining Tasks, Data Attributes and Measurement, Types of Data Sets, Measurement and Data Collection Issues, Data Preprocessing: Aggregation, Sampling, Dimensionality Reduction (5 Periods)

Unit 2: Basic techniques for Classification Decision Trees, Model Over fitting, Evaluating the Performance of a Classifier, Holdout Method, Random Subsampling, Cross-Validation, Bootstrap, Methods for Comparing Classifiers. (8 Periods)

Unit 3: Advanced Techniques for Classification Rule-Based Classifier, Nearest-Neighbor classifiers, Bayesian Classifiers, Artificial Neural Network(ANN), Support Vector Machine (SVM), Ensemble Methods: Bias-Variance Decomposition, Bagging, Boosting, The Receiver Operating Characteristic Curve (7 Periods)

Unit 4: Association Analysis: Basic Concepts and Algorithms Frequent Item set Generation-The apriori Principle, Rule Generation in Apriori Algorithm, Alternative Methods for Generating Frequent Item sets: FP-Growth Algorithm, Evaluation of Association Patterns, Objective Measures of Interestingness, Simpson's Paradox. (7 Periods)

Unit 5: Cluster Analysis: Basic Concepts and Algorithms The Basic K-means Algorithm, Agglomerative Hierarchical Clustering, The DBSCAN Algorithm, Strengths and Weaknesses of DBSCAN, Cluster Evaluation techniques. (7 Periods)

Unit 6: Cluster Analysis: Additional Issues and Algorithms Prototype-Based Clustering: Fuzzy Clustering, Clustering Using Mixture Models, Self-Organizing Maps (SOM), Density-Based Clustering: Grid-Based Clustering, Subspace Clustering, Graph-Based Clustering: Minimum Spanning Tree (MST) Clustering, Hierarchical Clustering with Dynamic Modeling, Scalable Clustering Algorithms (8 Periods)

Total: (42 Periods)

Reference Text:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Publishers, 2007, [Chap. 1,2, 4, 5.1-5.6, 6.1-6.6, 8, 9.1-9.4]

Suggested Readings:

2. Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann pub, 2001

3. Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann pub, 2011, 3rd Ed.

MMCS 3(P): Practicals: Mathematical Methods for Data Mining

(1 Credit)

Course objectives: This is a practical course intended to give coding level experience with regard to data mining subject.

Course outcomes: At the end of the course the student will be able to

- Implement some algorithms pertaining to clustering, classification, association finding, feature selection and visualization on real world data

Syllabus:

Relevant exercises from different units in the syllabus can be implemented in Lab.

Reference Text:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Publishers, 2007.